Multidisciplinary Modeling and Design of a Space Structure

THESIS

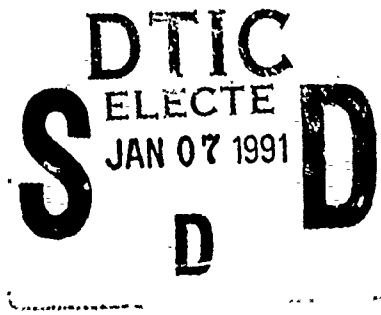
GSE-90D

AFIT/GSE/ENY/90D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENY/90D-1



Multidisciplinary Modeling and Design of a Space Structure

THESIS

GSE-90D

AFIT/GSE/ENY/90D-1

Approved for public release; distribution unlimited

AFIT/GSE/ENY/90D-1

Multidisciplinary Modeling and Design of a Space Structure

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Systems Engineering

Brian K. Cassiday
Captain, USAF

Stephen O. Gaines II
Captain, USAF

Lawrence L. Gatschet
Captain, USAF

Mario N. Moya
Captain, USAF

John T. Tester
Captain, USAF

December, 1990

Preface

This document is the result of the research undertaken by one group of students in the Systems Engineering Class of 1990 at the Air Force Institute of Technology (AFIT). We performed this research in the hope of furthering the state of the art in complex system modeling techniques. We feel this research will be very beneficial to the systems engineering design community.

This research could not have been completed without the guidance, expertise and support of many people whom we would like to thank. First of all, we would like to thank the personnel of the Weapons Lab at Kirtland AFB, NM for opening the space system design field to us. For their invaluable knowledge, guidance and time, we would like to thank our thesis committee: Maj Dave Robinson, Dr. Curtis Spenny, Dr. Brad Liebst, Capt Howard Gans and Maj Dave Stone. We would like to thank Dr. Dave Redding and Dr. Hugh Briggs of the Jet Propulsion Laboratory for providing us the basis of the optical ray trace program. Ms. Cathy Bush, Mr. Doug Burkholder and Mr. Jack Phillips deserve thanks for assisting us with computer hardware and software problems within AFIT. We thank Dr. Ronald Rosenberg at Michigan State University and his associates at Rosencode for help they provided allowing us to use the ENPORT software package. Finally, we would like to thank our families and friends for understanding our needs and providing the support that helped us complete this work. You all have our greatest thanks.

Brian, Steve, Larry, Mario, and John

GSE-90D

Table of Contents

List of Figures

List of Tables

Abstract

As man's presence in space grows, so does the size and complexity of the systems he deems necessary to put there. Unfortunately, man's ability to analyze several engineering disciplines interacting simultaneously has not improved. Current design techniques involve developing mathematical models of all subsystems to determine a performance level prior to actual production of the system. Normally these models are created separately and the interaction between the subsystems is not known completely as the design proceeds. Only when the system is finally built can the interactions between the subsystems be accurately defined. Current computer-aided design tools can be used to develop in-depth computer models of, at most, two disciplines simultaneously. This is important during the latter stages of a design but presents problems during the conceptual design phase.

A method is needed to simultaneously integrate several engineering disciplines into one complete model to determine the performance of the system as a whole during the design development. System engineering methodologies are particularly suited to this problem. These methods provide a structured approach to problem formulation and system parameter identification. Bond graphs are also well suited as bond graphs model power flow and energy relationships within subsystems, a characteristic inherent to all dynamic systems.

The intent of this research was to demonstrate the use of concurrent engineering theory to systematically model a large flexible space structure involving several engineering disciplines. The disciplines considered include: dynamics, controls, optics, structures, and heat transfer. A set of optimal solutions is presented and the results of this research are compared and contrasted with the results from classical design and modelling techniques.

Multidisciplinary Modeling and Design of a Space Structure

I. INTRODUCTION

1.1 BACKGROUND

As man's presence in space grows, so does the size and complexity of the systems he deems necessary to put there. Programs such as the Strategic Defense Initiative (SDI) and the NASA space station are generating stringent requirements for these large complex space systems. For example, an SDI satellite may be required to generate large amounts of energy and simultaneously track a moving target. The tracking is required to be accurate, but the energy generation process and attitude control inputs produce structural vibrations. These simultaneous events conflict and must be isolated from each other. These events are frequently isolated and controlled on earth; however, they present a unique and challenging problem when attempted in the space environment.

Working in space presents problems that are not normally encountered in man's own environment. Particular problems the space environment presents to the designer are reduced thermal dissipation and low gravity. Without the Earth's atmosphere, heat convection is impossible and the only method of heat transfer to the environment is radiation. The lack of environmental heat convection presents unique problems because extreme operating temperatures impact the design process. Low gravity causes multiple problems. Gravity helps provide a natural damping effect for vibrations by causing additional friction within the system itself. Another gravity related problem is the cost of lifting a mass out of Earth's atmosphere. The cost per pound for the launch of a satellite requires that the system be designed with the

minimum amount of weight possible. This results in flexible structures that increase the vibration and vibrational control problems.

Current design techniques involve developing mathematical models of all subsystems to determine a performance level prior to actual production of the system. Normally these models are created separately and the interaction between the subsystems is not known completely as the design proceeds. Only when the system is finally built can the interactions between the subsystems be accurately defined.

1.2 PROBLEM STATEMENT

A method is needed to simultaneously integrate several engineering disciplines into one complete model to determine the performance of the system as a whole during the design development. System engineering methodologies are particularly suited to this problem and bond graph theory appears to be the logical modeling approach. These methods provide a structured approach to problem formulation and system parameter identification. Identification of the significant system parameters is necessary if an accurate, complete, model of the system as a whole is to be achieved. The intent of this research is to demonstrate the use of systems engineering methods and bond graph theory to systematically model a large flexible space structure involving several engineering disciplines.

An underlying problem is that no software tool exists that allows simultaneous consideration of all the engineering disciplines involved in complex systems. Current software tools can be used to develop in-depth computer models of at most, two disciplines simultaneously. This is important during the latter stages of a design but presents problems during the project planning design phase.

The project planning design phase is used to determine the major interactions between different subsystems. The design engineer has difficulty determining important design variables at the lower design levels because of the numerous design variables present. In order to effect a sound design from the onset, a higher level

design tool is necessary to determine the major design variables present in the system. Thus the design engineer can base major design choices on the performance measures of the system utilizing the predominant design variables.

1.3 SUMMARY OF CURRENT KNOWLEDGE

There are many computer software packages available for engineering simulations, but there is no single software analysis package which combines several engineering disciplines into one system. The Jet Propulsion Laboratory (JPL) has developed an integrated control/structures/optics modeling method (Redding, D., 1989) and includes a software tool to aid in the analysis. This package is currently being used to design and analyze optical spacecraft (Briggs and others, No Date).

The Weapons Lab, located at Kirtland AFB, is presently developing an Integrated Structural Modeling (ISM) software package which will integrate multiple engineering software tools (AFWL, 1985). ISM will integrate mechanical structures, optics, controls, thermodynamics, and multibody dynamics. ISM will support general SDI simulation requirements as well as provide the capability to evaluate and analyze other complex structural systems. The Weapons Lab (WL/ARCB) also serves as the sponsor of the research described herein.

Stuctural analysis software tools are useful in both complex linear and nonlinear static and dynamic problems. Aeroelasticity, heat transfer and fluids analyses are additional capabilities of some of these tools; however, it is often impossible to directly link two or more of these capabilities into a single simulation. Furthermore, structural and optical simulation packages are always custom-made software developed by the users. As such, they have no generic interface with the commercial engineering community. The structural analysis in this research is mostly of a dynamic nature. Dynamic structural analysis is typically not interactive, but batch. In other words, the dynamic input to a system is predetermined and is not affected by the response of the structure. In a controlled structure, the controlled, dynamic

input is often a direct result of the structure's response or orientation. A structural analysis tool is needed which accepts a dynamic input from the output of another concurrently running simulation.

The automatic controls discipline plays a major part in many applications and has a large role on space platforms. Attitude control of space structures may be achieved in a variety of ways. One of the more popular methods for fine motion control is the use of momentum wheels. Many techniques are available for developing the dynamics models of structures (Martin, J., 1987). However, these techniques do not significantly address other engineering disciplines that may have an impact on the system dynamics. Therefore, the development of a precise control system involves methodical switching between dynamics models which can be time consuming and costly (AFWL, 1985).

Optical modeling involves formulating ray trace techniques to determine how the optics system will perform. The problem with this approach is the inability to interface the optics model with other models efficiently. Ray trace algorithms can be computationally inefficient and are cumbersome when trying to interject information from other systems. An efficient ray trace algorithm has been developed by JPL (Redding, D., 1989) that lessens the computation time. But a simple method to utilize the ray trace techniques within a simulation model is still lacking. Thus it is difficult to determine the effect structural displacements or control algorithm errors will have on the optics system.

The software packages in use today are generally useful for in-depth modeling of specific systems. Although these packages do have certain limitations, they are useful when dynamic interactions between neighboring systems are well known and can be accurately predicted. However, during project planning phases when particular subsystems are not accurately defined, dynamic interrelations are not well known, and particular performance parameters have not been set, using these software packages is cumbersome. The project planning phase requires a method to

quickly evaluate different design options based on performance and dynamic interactions. The ability to perform this function is severely lacking and requires new methods for accomplishing the goals of the project planning design phase, identifying the major design parameters of the system.

1.4 SCOPE

The research was carried out in different steps in order to accomplish the overall goals. Systems engineering techniques were used to simultaneously analyze all the major engineering disciplines involved. Computer models of the system, using bond graphs and traditional modeling techniques, were developed for analysis. The design process was completed through the initial design phase only. When completed, these models served as the demonstration of the integration method.

The key issues in the design and analysis of the space system were the determination of the major engineering disciplines involved, the identification of critical system parameters, the identification of parameter interactions, and the determination of system performance. The project was limited to the determination of these key issues. A limited optimization of the design was conducted, as the primary purpose was a demonstration of the tools and techniques and not to design a satellite for NASA.

1.5 APPROACH/METHODOLOGY

Bond graphs were utilized to the largest extent possible to model the system. Bond graphs are unique in that they are only concerned with power flow within a system, a property inherent to all engineering disciplines. Because bond graphs have not been utilized on systems as complex as a space structure, traditional engineering techniques will also be used to verify results. The research was conducted in three steps.

Step 1 involved the initial system characterization. In this step an appropriate space system was selected for design. The system specifications were selected based on discipline interaction and complexity under the assumption that the system was not intended to have a real world application.

Step 2 involved system design and analysis using system engineering techniques. This step represents the project planning phase of Hall's morphology box (Hall, A., 1969) and also represents the largest portion of the research. Within this step the techniques of bond graphs were applied in an attempt to create top level models of the system chosen in step 1. The major disciplines and significant system elements to be modeled were chosen during this phase. Interaction between the different disciplines was defined and the significant system parameters identified. Finally, a limited optimization of the system was conducted.

Step 3 provided a comparison/contrast of the techniques used in the methods being researched here and those of traditional methods. This serves as the results of this research and is intended to point out the advantages and disadvantages of each.

The research results presented here are arranged such that distinctions between the different steps is recognizable. In addition, five logic steps of the systems engineering process can also be recognized in step 2.

II. SYSTEM CHARACTERIZATION

In order to accomplish the goals of this research, a system was defined upon which the modeling methods could be applied. The reader is cautioned to remember the objective of this research: demonstration of multidisciplinary modeling techniques. The resulting system design is used only as a vehicle to demonstrate the techniques.

The system chosen was based upon the desires of the sponsor (WL/ARCB), the relative complexity of the system, and the different disciplines involved. These engineering disciplines were selected to relate to the design problems specifically encountered by the Weapons Lab. In recent years, the Weapons Lab has performed considerable research on SDI and the design of related satellites. Thus, a satellite system was selected. The team concentrated on engineering specialties dealing with flexible structural dynamics, controls, and optical design.

A search for a specific satellite design problem began. A satellite concerned with some aspect of national defense was desirable. The destructive satellites frequently associated with SDI are far too complex to consider: satellites which detect and report 'targets,' or ICBMs, were considered instead. They are essentially reconnaissance satellites, with the additional capability of continuously changing their attitude to center attention on a moving target. The scenario in which this satellite is used was developed:

- A minimum line of sight error
- It must be carried into orbit by the space shuttle
- It must be able to detect a missile plume
- It must be able to track the trajectory of an ICBM
- It is in orbit over a suspect land mass.

- It uses its attitude controllers to center its line-of-sight (LOS) on the missile.
- It transmits the collected data to earth as it tracks the target.

Because of the unique nature of this research, the selection of initial design was not based on creating a highly efficient tracking satellite. Instead, emphasis was placed on selecting systems that were easily visualized and could be simply evaluated. The intent was to select a system, provide performance objectives on which to base decisions, and develop the system for analysis.

III. SYSTEM DESIGN AND ANALYSIS

In applying system engineering techniques to design and analyze the system, five logic steps of the Hall's activity matrix (Hall, A., 1969) were carried out for the project planning phase. The five logic steps are:

1. Problem definition
2. Value system design
3. System synthesis
4. System analysis and modeling
5. Optimization

3.1 PROBLEM DEFINITION

The problem definition setup is aimed at developing a thorough understanding of the problem. The main activities of this step were the identification of the system needs, constraints, and alterables and then identification of thier interactions.

The system needs, constraints, and alterables were developed based on the system characterization in step 1. They address the overall system characteristics or requirements and were constant throughout the research.

The system needs are those characteristics required or desired of the system. They are a restatement of the system or design requirements. The system constraints identify bounds on how the needs may be satisfied. They are used to qualify alternative solutions or designs as feasible or non feasible; they place limits on the design. The system alterables define the scope of the system and design options. Table 3.1 shows the system needs, constraints, and alterables that were developed. Interactions between the system needs, constraints, and alterables are shown in Figure 3.1.

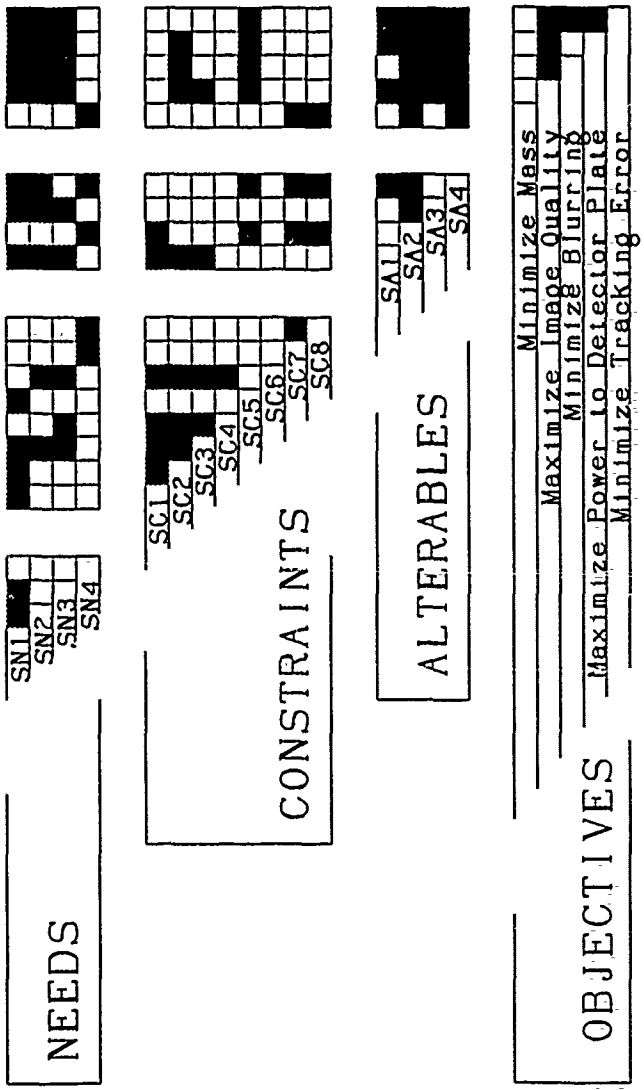


Figure 3.1. System level needs, constraints, and alterables interactions

Table 3.1. System needs, constraints, and alterables

Needs	SN 1.	Clear image of missile on the detector plate
	SN 2.	Acquire the missile
	SN 3.	Track the missile
	SN 4.	Be designed with minimum mass
Constraints	SC 1.	Zero degree inclination orbit
	SC 2.	Circular orbit
	SC 3.	Missile trajectory must be accessible for viewing
	SC 4.	Missile flight is that of an ICBM launched from the USSR
	SC 5.	Optical system is a cassegrainian telescope
	SC 6.	Initial position of system in inertial space is known
	SC 7.	Maximum weight of the system is 52,000 pounds
	SC 8.	Maximum size is confined to a cylinder with radius 15 feet and length of 60 feet
Alterables	SA 1.	Radius of the orbit
	SA 2.	Structural configuration of the system
	SA 3.	Controller
	SA 4.	Size of optics system

3.2 VALUE SYSTEM

In applying system engineering methods, a critical step in the problem solving procedure is the development of a value system. A value system is defined as the set of objectives to be met in resolving the problem and a set of performance measures on the objectives by which to determine their attainment (Hill and Warfield, 1972). The objectives are developed based on the needs, constraints, and alterables defined in the problem definition step.

3.2.1 Objectives The primary objective of this research is to design and model a large flexible space structure involving several engineering disciplines. To support this, a secondary objective was selected: Design an intercontinental ballistic missile (ICBM) imaging satellite. Based on this objective and the set of system needs listed in Table 3.1, the following subobjectives were established:

1. Minimize mass

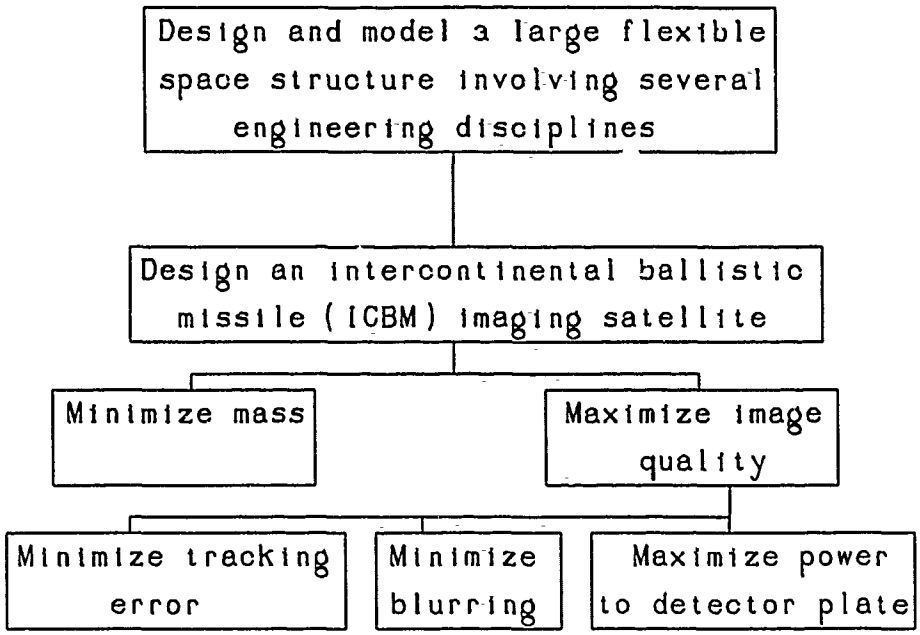


Figure 3.2. Objective tree

2. Maximize image quality
3. Minimize blurring
4. Maximize power to detector plate
5. Minimize tracking error

The subobjectives were selected to differentiate among the design alternatives since the two top objectives are not easily quantifiable. The top level objectives along with their subordinate subobjectives are arranged in a hierarchical tree as shown in Figure 3.2. Interactions between the subobjectives and the system needs, constraints, and alterables are shown in Figure 3.1. Each of the five subobjectives will be explained separately.

Minimize Mass Due to the extreme cost of placing systems in orbit, minimizing mass is an objective for every space system. The maximum system mass is constrained by the capacity of the space shuttle. The total mass of the system can be used to directly measure this objective.

Maximize Image Quality In designing an imaging satellite, a primary objective is to maximize the image quality. Although it is possible to measure this objective directly, from a design standpoint, it is more effective to evaluate the three subordinate objectives - minimize tracking error, minimize blurring, and maximize power to the detector plate.

Minimize Tracking Error Tracking error (line- of-sight error) is the angle between the optical axis orientation and that of the vector from the primary mirror to the missile. Tracking error relates directly to the ability of the control system to track the missile. It also effects image quality. As the missile moves off the optical axis, off-axis aberration increases and image quality decreases. The tracking error can be measured directly on the detector plate as the distance between the image center and the center of the detector plate.

Minimize Blurring Blurring of the image on the detector plate at a given instant in time is the result of two phenomenon - defocus and optical aberrations. Defocusing is a factor because the optical model was designed with a fixed focus. As a consequence, the missile image will be out of focus if the missile is at a distance other than that for which the system is focused. Optical aberrations are a cause of blurring in every optical system. These aberrations are explained in detail in the optics model section. The main consideration is that as the missile moves off-axis, the effect of optical aberrations increase. Thus, blurring and tracking error are highly correlated. Blurring of the image on the detector plate over time is the result of the above mentioned phenomenon plus a change in the tracking error. The tracking error changes over time due to missile movement, controller commands, and vibrations. Blurring can be measured in the model as the variance over time of the individual ray traces on the image plate. The term variance often implies a statistical randomness associated with collected data. Such is not the case in the simulation, for all of the trace outputs are completely deterministic. However, since the detector image is in

the form of scattered points, each group is defined as having a spatial mean and variance.

Maximize Power to the Detector Plate The objective to maximize the light intensity from the missile that is received by the system is based on the assumption that the more light received, the better the image. The light striking the primary mirror can be measured based on the initial intensity of the source, the distance from the source to the system, and the area subtended by the primary mirror. Not all the light striking the primary mirror is guaranteed to reach the detector plate. In the optical ray tracing model, some of the rays may miss the secondary mirror, not pass through the primary mirror hole, miss the lens, or miss the detector plate. The number of rays reaching the detector plate is measured in the optical model. Therefore, the percentage of light transmitted from the primary mirror to the detector plate can be determined.

3.2.2 Performance Measures The final component of the value system is the performance measures. A performance measure is used to determine how well an objective has been satisfied and must be meaningful and measurable. The only objectives requiring performance measures are those at the lower level. This is due to the fact that objectives are broken down into subobjectives to reach a level of detail where the objective is measurable. The performance measures selected are:

1. System mass
2. Optical system mean tracking error
3. Inverse of the light intensity upon the detector
4. Light intensity at the detector divided by the variance

Interactions between the subobjectives and the performance measures are shown in Figure 3.3.

Figure 3.3. Subobjective and performance measure interactions

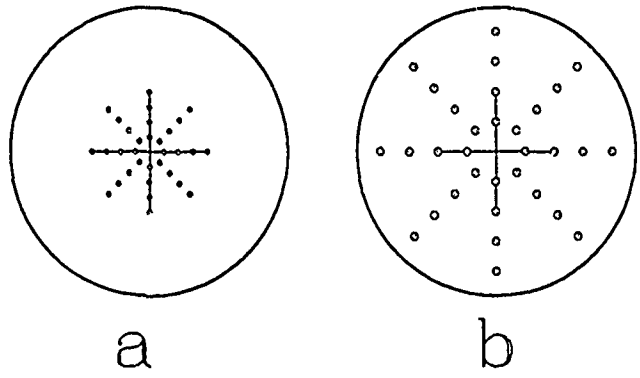


Figure 3.4. Power vs. variance illustration.

The only performance measures not yet discussed is the optical system light intensity at the detector divided by the variance. In communication theory a frequently used measurement is the 'signal-to-noise ratio.' The optical simulation developed can combine the light intensity and the image spatial variance into a similar measurement, the "intensity-to-variance ratio" (IVR). The concept behind the IVR is illustrated by Figure 3.4.

Both Figures 3.4 *a* and 3.4 *b* show hypothetical detector images. However, the first image is a tighter image than that of the second. If the intensity measurement for both were the same, then the variance of Figure 3.4 *a* would best satisfy the top objective. But if the intensity measurement of Figure 3.4 *b* were significantly greater than that of Figure 3.4 *a*, then selecting the best image to satisfy both subobjectives of objective one would be difficult. A solution to this problem is the IVR.

3.3 SYSTEM SYNTHESIS

The system synthesis step is aimed at conceptualizing potential designs which allow attainment of the objectives. The product of this step is a series of activities which form a plan for evaluating the designs with respect to attaining the objectives.

Because of the nature of this research, only one design was selected. The design was based on the system characterization of step 1 and the system needs, constraints,

and alterables identified during problem definition. The following activities were then generated:

1. Determine interactions between subsystems
2. Develop and verify subsystem models
3. Integrate subsystem models
4. Optimize the design

The subsystems were identified in the system characterization step as controls, structures, and optics.

At this point, the problem definition was reevaluated. The problem definition step was now aimed at identification of design level needs, constraints, and alterables.

The design level needs, constraints, and alterables define specific characteristics or parameters of each subsystem. They were developed based on extending the system needs, constraints, and alterables down to the subsystem design level. The design level needs, constraints, and alterables were not constant throughout the research. They evolved as the design evolved.

The design level needs are specific characteristics or parameters required for use within the system models. The needs of one model must be obtained as an input from outside the system or generated by another model. The design level needs identified for the subsystem models are listed in Table 3.2.

The design level constraints are specific limits placed on each model. The design level constraints are listed in Table 3.3.

The design level alterables are the parameters within each model which can be improved, changed or deleted as the design evolves. The design level alterables are listed in Table 3.4.

Interactions between the design level needs, constraints, and alterables are shown in Figure 3.5.

Table 3.2. Design level needs

MODEL	ID No.	NEED
Controls	DN 1	Azimuth and elevation of structure
	DN 2	Missile representation on detector plate
	DN 3	Inertia information about principle axes
	DN 4	Orientation of momentum wheels
		WRT principle axes
Structures	DN 5.	Distance between primary mirror and detector
	DN 6	Primary mirror diameter
	DN 7	Detector plate diameter
	DN 8	Optical system mass
	DN 9	Optical system inertia tensor
	DN 10	Momentum wheel masses (3)
	DN 11	Momentum wheel inertia tensors (3)
Optics	DN 12	Free line of sight
	DN 13	Coordinates of the missile in mirror frame

Table 3.3. Design level constraints

MODEL	ID No.	CONSTRAINTS
Controls	DC 1	Momentum wheels are primary controllers
	DC 2	Fixed momentum wheel density
Structure	DC 3	Tetrahedron structure
	DC 4	Solid circular cross-sectional beams
	DC 5	Aluminum or graphite composite beams
	DC 6	Welded beam joints
	DC 7	Welded optical system/structure joints
	DC 8	Structure contains the primary mirror and detector
	DC 9	Side beams are of same length
	DC 10	All beams are of same material
	DC 11	All beams are of same cross-sectional size
Optics	DC 12	Optical structure is rigid
	DC 13	Honeycomb mirror structure

Table 3.4. Design level alterables

MODEL	ID No.	ALTERABLES
Controls	DA 1	Mass of controllers
	DA 2	Modes to be controlled
	DA 3	Settling time
	DA 4	Overshoot
	DA 5	Algorithm
Structure	DA 6	Beam materials
	DA 7	Side beam length
	DA 8	Beam diameter
	DA 9	Modal damping coefficient
	DA 10	Controller placements
Optics	DA 11	Focal length of primary mirror
	DA 12	Focal length of secondary mirror
	DA 13	Focal length of lens
	DA 14	Eccentricity of secondary mirror
	DA 15	Diameter of primary mirror
	DA 16	Diameter of secondary mirror
	DA 17	Diameter of lens
	DA 18	Diameter of detector plate
	DA 19	Hole diameter in primary mirror
	DA 20	Detector resolution
	DA 21	Distance between primary and secondary mirrors
	DA 22	Distance between secondary mirror and lens
	DA 23	Distance between lens and detector plate

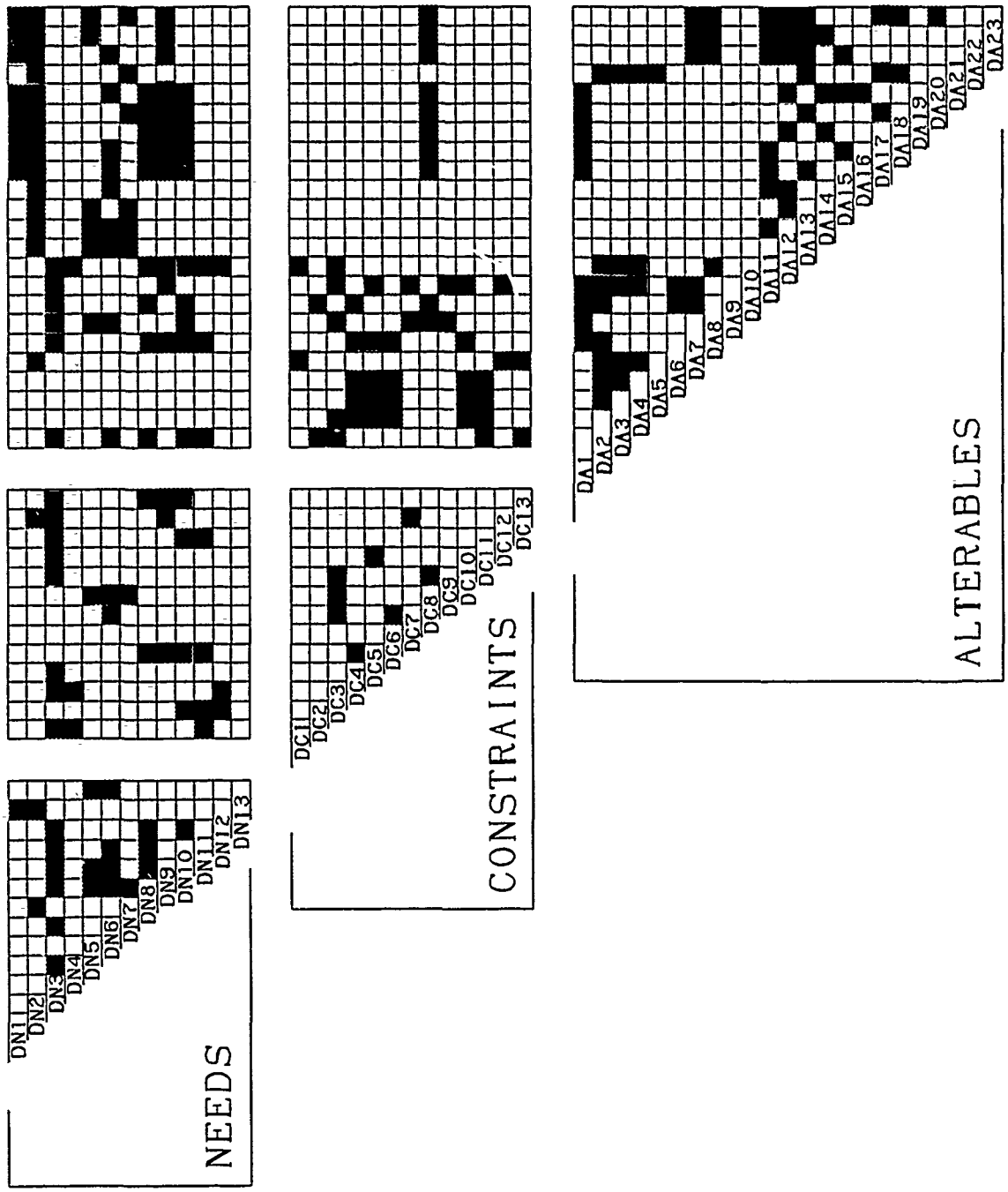


Figure 3.5. Design level needs constraints and alterables interaction

Interactions between the subobjectives and the design level needs, constraints, and alterables are shown in Figure 3.6.

3.4 SYSTEM ANALYSIS AND MODELLING

In order to accomplish the goals of this research, computer models of the system being designed had to be constructed. The models can be broken down into six basic groups: missile dynamics, orbital dynamics, controls, structural vibrations, optics, and rigid body rotations. These models were originally developed using either bond graphs or other techniques that were then converted to bond graphs. The optics model, however, was developed completely as a FORTRAN subroutine. Each model was specifically developed to provide the input and output parameters necessary to integrate with the rest of the system.

The missile flight dynamics model is based upon predicted flight characteristics of an ICBM. The model provides flight information on the missile with respect to a reference frame attached to the center of the earth. The orbital dynamics model shows the orbit characteristics of the satellite and is used in conjunction with the missile model to provide position information of the missile with respect to the system.

The control model is a closed loop system that uses the actual plant representing the structure. The control system receives commanded position information from the controlling computer and is based either on optical information or targeting information from another system. In either case, the control system utilizes current position information from an onboard Inertial Navigation Unit (INU). The INU is simulated in the model as a FORTRAN program that uses information already available elsewhere in the model to calculate current position with respect to an inertial frame.

The structural vibrations model takes information derived from a NASTRAN analysis of the structure and implements it in bond graphs. The bond graph repre-

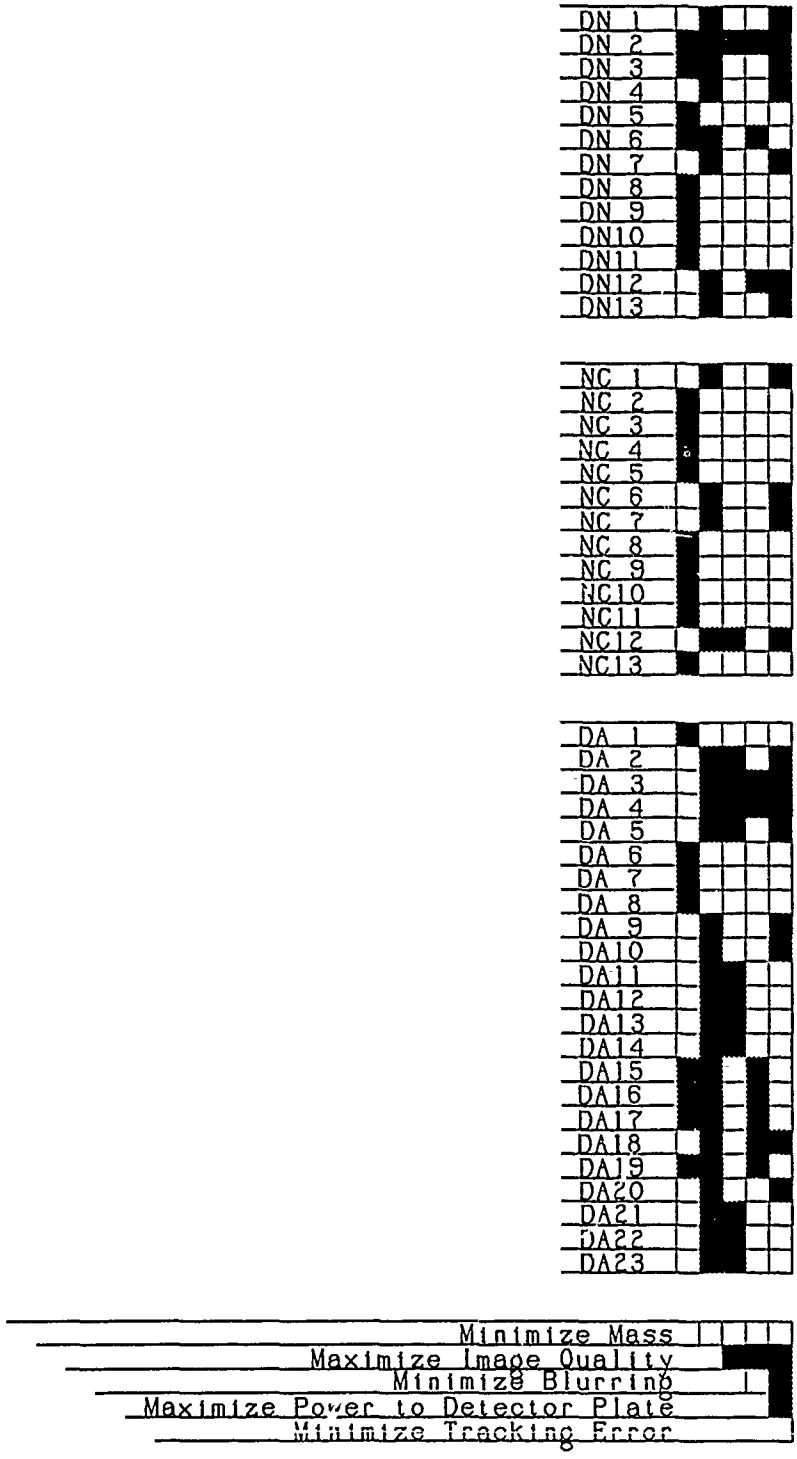


Figure 3.6. Subobjective interaction at the design level

sentation takes force inputs to the system, modifies them through a series of transformations and gives vibrational motions at the points of interest. For this system, the points of interest are at the mirror attachment points on the structure.

The optics model is a FORTRAN subroutine utilizing exact ray trace methods to calculate a target position on the optical image plate. This model receives as input a vector telling the optics the exact position of the missile with respect to the primary mirror. The exact ray trace technique also shows aberrations in the image due to off-axis positions and natural optical component phenomenon. Since the missile typically shows up on the image plate with some degree of aberration, the position of the missile on the image plate is output as a mean value of the distribution and the variance (or blurriness) of the image.

3.4.1 Missile Dynamics The missile dynamics model is used to provide inputs to the system model during performance evaluation. The model simulates the launch of a two stage Titan III missile from Kansk, USSR. The trajectory is that of an ICBM with a target destination of Dayton, OH. The simulation starts at launch and ends at second stage burnout.

3.4.1.1 Physics The missile model is based on the principle of two body dynamics. The model was developed under the assumption that only two forces act on the missile: gravity and thrust. An in-depth derivation of the missile dynamics can be found in Appendix A. Because the mass of the missile is not constant, it is more convenient to model accelerations. The missile acceleration due to gravity is:

$$A_g = \frac{GM}{R^2} = \frac{\mu_e}{R^2} \tag{3.1}$$

where,

M = Mass of earth

G = Gravitational constant

μ_e = GM

R = Distance of satellite from center of earth

The acceleration due to thrust is:

$$A_T = \frac{THRUST}{TOTAL_M - MFR \cdot TIME_S} \tag{3.2}$$

where,

$THRUST$ = Missile thrust

$TOTAL_M$ = Total mass of the missile

MFR = Mass flow rate of the stage

$TIME_S$ = Time of stage burn

The missile acceleration is then modeled in three dimensions as:

$$\dot{V}_X = \frac{-\mu_e R_X}{R^3} + \frac{THRUST}{TOTAL_M - MFR \cdot TIME_S} \cdot \frac{V_X}{V} \tag{3.3}$$

$$\dot{V}_Y = \frac{-\mu_e R_Y}{R^3} + \frac{THRUST}{TOTAL_M - MFR \cdot TIME_S} \cdot \frac{V_Y}{V} \tag{3.4}$$

$$\dot{V}_Z = \frac{-\mu_e R_Z}{R^3} + \frac{THRUST}{TOTAL_M - MFR \cdot TIME_S} \cdot \frac{V_Z}{V} \tag{3.5}$$

The parameters R_X, R_Y, and R_Z are the X, Y, and Z position of the missile relative to the earth's center in earth frame coordinates. V_X, V_Y, and V_Z are the missile velocities in the X, Y, and Z earth frame coordinates.

3.4.1.2 Interface The interface between the missile dynamics model and the system model is minimal. The missile dynamics model passes three pa-

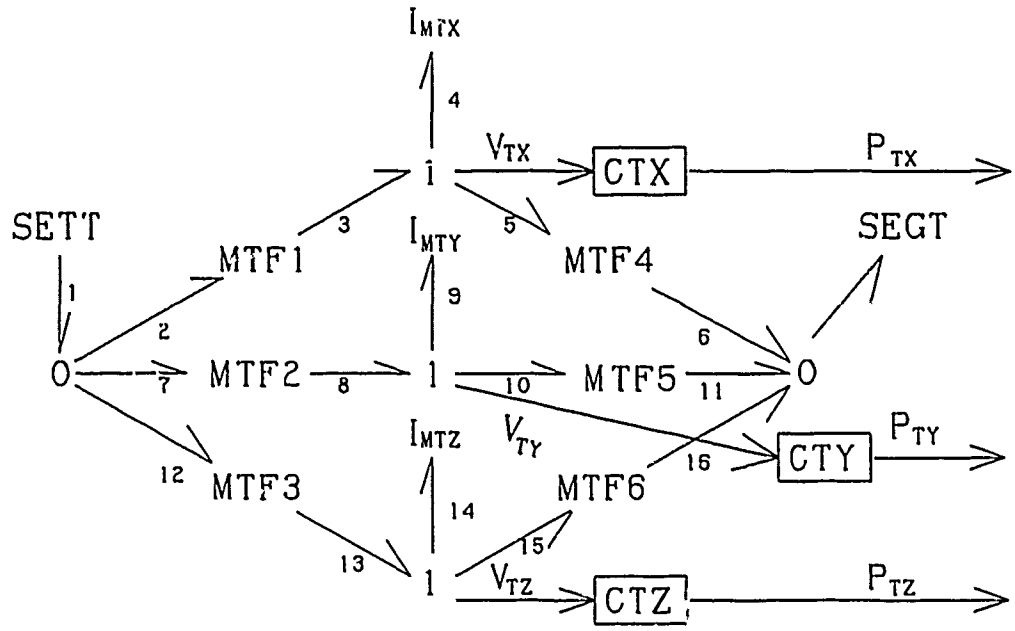


Figure 3.7. Missile dynamics bond graph

rameters to the system model. The parameters are the X, Y, and Z position of the missile relative to the earth's center in earth frame coordinates. There are no parameters passed from the other system models to the missile model.

3.4.1.3 Missile Bond Graph Description The bond graph of the missile dynamics is shown in Figure 3.7 (signal bonds for the modulated transformers are not shown). The missile is modeled as a unit mass, in three dimensions, driven by two acceleration sources. Source SETT is acceleration due to thrust and SEGT is acceleration due to gravity. Both SETT and SEGT change throughout the simulation. SETT is a function of both the stage parameters and the total mass of the missile. The total mass of the missile changes throughout the simulation due to propellant loss and stage separation. SEGT is a function of the distance of the missile from the center of the earth.

Normally, bond graphs model sources as force sources and thus model power flow within the system. Because the mass of the missile changes over time, it was

not possible to model power flow with the bond graph software used. Also, because there is no power flow between the missile dynamics model and the other system models, it was necessary to model only the missile kinematics. By using acceleration sources as apposed to force sources, the missile dynamics model is truly a missile kinematics model.

The acceleration of the missile due to thrust is modeled as the dependent source SETT where:

$$SETT = \frac{THRUST}{TOTAL_M - MFR \cdot TIME_S} \tag{3.6}$$

The acceleration due to gravity is modeled as the dependent source SEGT where:

$$SEGT = \frac{\mu_e}{R^2} \tag{3.7}$$

and,

$$R = \sqrt{P_{TXE}^2 + P_{TYE}^2 + P_{TZE}^2} \tag{3.8}$$

Modulated transformers MTF1, MTF2 and MTF3 are functions of velocity and are modeled as:

$$e_3 = e_2 \cdot \frac{V_{TX}}{V} \tag{3.9}$$

$$e_8 = e_7 \cdot \frac{V_{TY}}{V} \tag{3.10}$$

$$e_{13} = e_{12} \cdot \frac{V_{TZ}}{V} \tag{3.11}$$

where,

$$V = \sqrt{V_{TX}^2 + V_{TY}^2 + V_{TZ}^2} \tag{3.12}$$

Modulated transformers MTF4, MTF5 and MTF6 are functions of position and are modeled as:

$$e_5 = e_6 \cdot \frac{P_{TXE}}{R} \tag{3.13}$$

$$e_{10} = e_{11} \cdot \frac{P_{TYE}}{R} \tag{3.14}$$

$$e_{15} = e_{16} \cdot \frac{P_{TZE}}{R} \tag{3.15}$$

V_{TX}, V_{TY}, and V_{TZ} are the missile velocities in the X, Y, and Z earth frame coordinates. The velocities are passed, using signal bonds, to blocks CTX, CTY and CTZ which are block diagram integrators The signals P_{TXE}, P_{TYE}, and P_{TZE} are the desired X, Y, and Z position of the missile in earth frame coordinates.

The trajectory of the missile is determined by the initial position and initial velocity. The initial position is established by the initial condition of the three integrators. The initial velocity is established by the initial velocity (momentum) given to the masses I_{MTX}, I_{MTY}, and I_{MTZ}.

The initial conditions were adjusted until the desired trajectory and velocities were obtained. This desired (standard) trajectory was used during all future simulations.

3.4.2 System Orbital Dynamics The orbital dynamics model is used to provide the position of the satellite in inertial space. The model is flexible enough to

simulate both geosynchronous and non-geosynchronous orbits. The orbit is determined by the initial conditions given to the model.

3.4.2.1 Physics The orbital model is based on the principle of two body orbital dynamics. The model was developed under the assumption that the only force acting on the satellite is gravity. The force due to gravity is:

$$F = \frac{GMm}{R^2} = \frac{\mu_e m}{R^2} \tag{3.16}$$

where, M, G, μ_e, R are as previously defined, and m is the mass of the satellite.

3.4.2.2 Interface The orbital dynamics model passes three parameters to the rest of the system. The parameters are the X, Y and Z position of the satellite relative to the center of the earth expressed in inertial coordinates. There are no parameters passed from the other system models to the orbital dynamics model.

3.4.2.3 Orbital Dynamics Bond Graph The bond graph of the orbital dynamics is shown in Figure 3.8 (signal bonds to the modulated transformers are not shown). The satellite is modeled in three dimensions with a single effort source. The source SEGS models gravity as:

$$SEGS = \frac{\mu_e m}{R^2} \tag{3.17}$$

where,

$$R = \sqrt{P_{SX}^2 + P_{SY}^2 + P_{SZ}^2} \tag{3.18}$$

Modulated transformers MTF7, MTF8 and MTF9 are functions of position and are modeled as:

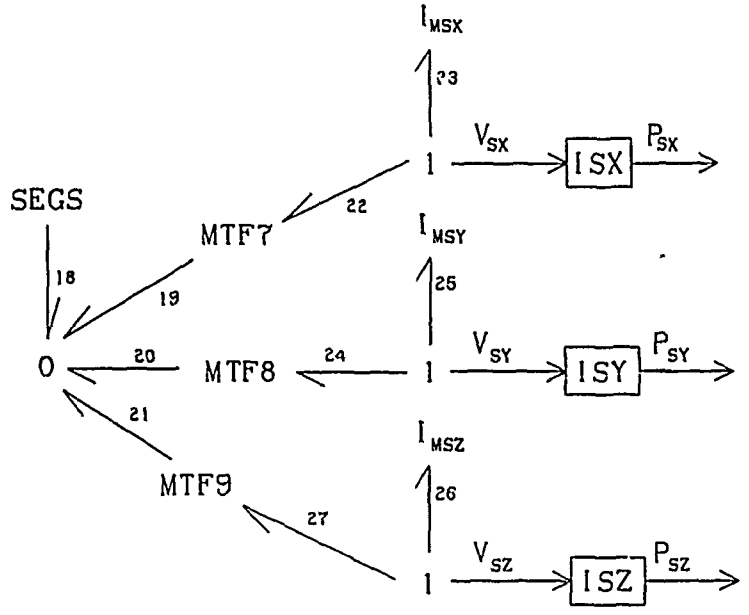


Figure 3.8. Orbital dynamics bond graph.

$$e_{22} = e_{19} \cdot \frac{P_{SX}}{R} \tag{3.19}$$

$$e_{24} = e_{20} \cdot \frac{P_{SY}}{R} \tag{3.20}$$

$$e_{27} = e_{21} \cdot \frac{P_{SZ}}{R} \tag{3.21}$$

V_{SX}, V_{SY}, and V_{SZ} are the X, Y, and Z velocity components of the satellite in inertial coordinates. The velocities are passed, using signal bonds, to blocks ISX, ISY and ISZ which are integrators. The signals P_{SX}, P_{SY}, and P_{SZ} are the desired X, Y, and Z position components of the satellite in inertial coordinates.The orbit is determined based on the initial conditions given to the model which must be specified for position (initial condition of the integrators) and the initial momentum of the masses.

3.4.3 Controls The satellite attitude was designed to be changeable with the motion created by a controlling algorithm utilizing desired position information. Momentum wheels provide the torque input to the system to cause rotation in a desired direction. The desired position information can come in three different forms.

The first form is a desired zero position, the position the satellite takes when no target information is available. The second form is target position as provided by another system. This information is passed to the system from a separate system that has detected launch and is handing off the tracking responsibilities. The third form is the optical position information. This is the position of the target as measured by the on board optics system.

The optical position information is the most crucial to the systems success. This is the only mode by which the satellite can provide any useful information to personnel on the ground or other tracking systems. As will be seen later, control via the optical information provides unique problems with nontrivial solutions.

3.4.3.1 Bond Graph The basic bond graph of a single momentum wheel is shown in Figure 3.9. A momentum wheel uses a high rotational momentum wheel mounted on a bearing and axle which is rigidly attached to the structure. A torque motor mounted on the axle provides a torque input to the wheel which in turn reacts and the input torque is then reflected into the structure thus causing an induced motion. In other words, the structure acts as the base from which the torque motor uses to push against to turn the momentum wheel. The associated bond graph shows a torque loss due to damping between the wheel and the structure (R) and even distributions of the net torque on the structure inertia (JS) and the wheel inertia (JW). As the bond graph indicates, the induced velocities in the wheel and the structure are in opposite directions.

As shown, the bond graph represents a two state system with the inertial momentums of the wheel and structure being the state variables. The bond graph

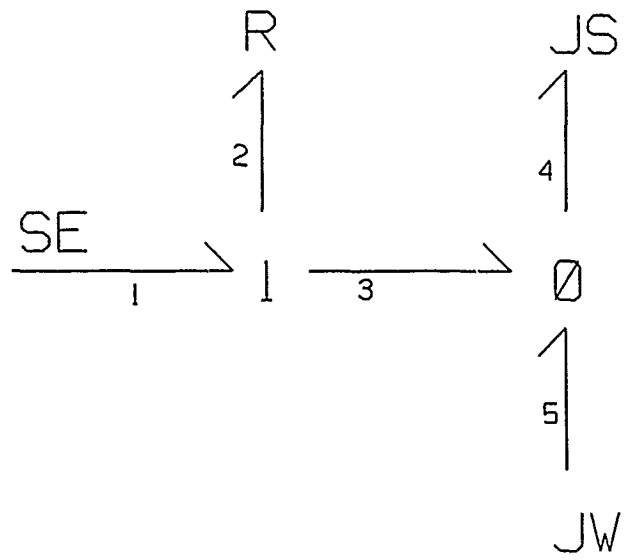


Figure 3.9. Momentum wheel bond graph

provides a simple method for representing the physical dynamics of the momentum wheel structure.

Although the equations for this system could certainly have been determined using other techniques, bond graphs utilize energy transfer relationships to determine the dynamical equations of the system. This is similar to Newton-Euler formulations but provides a graphical technique for showing the energy relationships within the system. This graphical representation can provide unique insights into a system and shows exactly how different systems interface together and energy is transferred between them.

After performing algebraic manipulations on the bond graph system equations (as shown in Appendix D) the state equation matrices were formed. Applying algebraic manipulations, the resulting transfer function from the torque input to the induced structure rotational velocity is represented by:

$$GTF = \frac{\frac{1}{J_s}}{S + \frac{R(J_w - J_s)}{J_w J_s}} \tag{3.22}$$

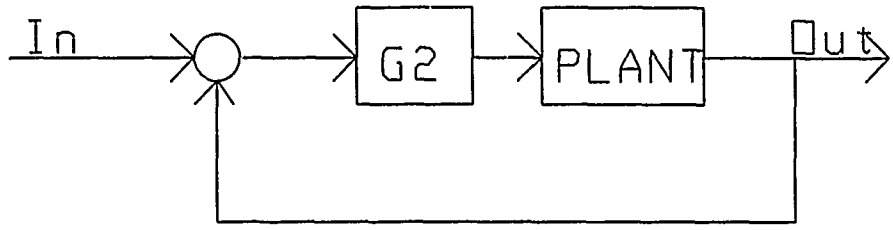


Figure 3.10. Attitude controller block diagram

This represents an inherently unstable system that must be controlled through feedback loops and compensators to provide desired time response characteristics.

Further investigation of the physical system reveals the damping factor, R, is considered to be so small as to be negligible. Since R is the coefficient of friction of the bearing between the momentum wheel and the structure, the assumption that the value of R is very small should be valid. As such, the pole in the right half plane of equation 3.22 is nearly at the origin. To simplify the model and reduce the number of states, the damping is considered to be zero. The transfer function is then reduced to a simple integrator with the inverse of the structure inertia as the gain component:

$$GTF = \frac{\frac{1}{J_s}}{S} \tag{3.23}$$

The intent of the design was to provide a control system that produces the desired platform rotation in a minimum of time without creating excessive vibrations in the structure. Therefore, the design was based on developing a compensator to give a particular overshoot and a particular settling time. The attitude control system block diagram is shown in Figure 3.10. This figure shows the compensator G2 and the plant of Equation 3.23. Different settling times were investigated during the optimization portion of the research while keeping the overshoot the same.

The compensator design is based on root locus techniques. As seen in Equation 3.23, the open loop transfer function of the plant essentially has two poles at

the origin when an integrator is added. The integrator is used to provide position information instead of velocity. The desired roots were found by choosing a desired overshoot and a desired settling time. Given the magnitude of the system response the first time it overshoots the final value is M_p, the system parameters that result in this value are:

$$M_p = 1 + \exp \frac{-\zeta\pi}{\sqrt{1-\zeta^2}} \tag{3.24}$$

Solving for ζ gives the desired damping ratio of the final closed loop system based on a simple second order response. With ζ known and given a desired settling time, the desired natural frequency of the second order response may be found via:

$$T_s = \frac{4}{\zeta\omega_n} \tag{3.25}$$

The desired roots of the characteristic equation are then given by:

$$\sigma = \omega_n\zeta \tag{3.26}$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \tag{3.27}$$

where,

ω_d = desired root imaginary part

σ = desired root real part

The final trick in this derivation is to select compensator poles and zeros to give the desired effect. In reality, there are an infinite number of choices that will yield the desired roots. However, the objective is to select roots that not only yield the desired closed loop roots but also make the other roots of the characteristic equation non-dominant. A root is considered non- dominant if the real part of the root is at least six times to the left of the real part of the dominant root. If not all roots, excluding the desired roots, are non-dominant, a simple second order response is not

achieved. It is then up to the design engineer to either move the compensator roots or decide if the resulting time response characteristics are satisfactory.

3.4.3.2 Interface with Structure As seen in Equation 3.23, the basic plant gain is highly dependent on the value of the structure rotational inertia. The basic control design dictates control of motion about each of the three principle axes. Unfortunately, the nominal plant depicted in Equation 3.23 is not constant. This is due to the cross coupling terms involved in Euler's equations for rotating rigid bodies (Likins, P., 1973). However, since the rotation about any given axis is considered to be small, the cross coupling terms are also small and the nominal plant is chosen to represent the system being controlled. Therefore, the design of the basic control system requires the value of the rotational inertia *a priori.*

In interfacing with the structure, the control system applies a control torque to the structure at specified locations. This torque is applied directly to one of the principle rotational inertia nodes. However, the control input will also induce vibrations into the system. Therefore, the interface between the controls system and the structure must also contain a path to feed torque information to the structural vibrations portion of the system bond graph. This is accomplished by pulling a velocity signal off of the controls input node and feeding that signal to the structural vibrations transformation section. Thus, the control system provides the torque inputs to the structure to control the structures orientation and also provides the necessary information to simulate the vibrations that should be seen in the structure as a result of the torque input.

3.4.3.3 Results The control system was first tested separately on a simplified system to ensure the system performed as expected. The simple system performed very well and was then integrated into the structure. The control system became unstable when integrated with the entire structural model as a result of induced vibrations in the structure.

As mentioned, the control system design was based on a nominal plant. This plant does not reflect any of the vibrations expected in the structure. Because the optical system is directly coupled to the structure, the optical system 'sees' all the vibrations in the structure with only structural damping available to remove the vibrations. With each vibration, the control system reacts. As the control system reacts, the vibrations in the structure increase to the point the system becomes unstable. This is a well documented problem that does not necessarily have a simple solution (Millar, R.A., 1977). For low flexible mode frequencies, the controller interacts unfavourably with the flexible structure and system stability is lost.

The first attempt at relieving the problem was to put a prefilter on the compensator of the control system to eliminate the vibrations. This had limited success. The filter was able to slow down the process but was unable to completely remove the vibrational information from the control system. Eventually, the system became unstable again.

The solution, for this research, was to eliminate the vibrations from the control system by controlling the system based on rigid body motion only, effectively eliminating the optical system from the control system feedback loop. This is comparable to receiving tracking data from another system and then utilize inertial navigation unit information to point in the desired direction. Documented solutions to this problem reflect a similar approach whereby gross motion control is utilized until reasonable pointing accuracy is achieved. The vibrations are damped out and then a sliding mode control algorithm is utilized, along with prefilters, to provide the fine motion control (Millar, R.A., 1977). This problem is in an area of active research, but the objective of the research presented here was to demonstrate a technique, not to solve the vibrations problem.

The final problem to be resolved was limiting the steady state error in the control system. As the missile moves through the sky, the missile is accelerating. In order for a tracking controller to track an accelerating target, the system type

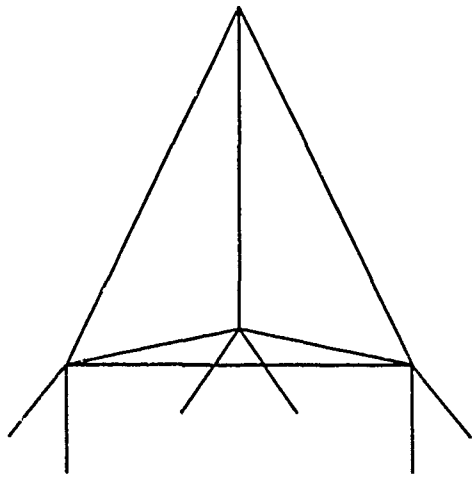


Figure 3.11. CSDL model 1.

must be at least 3 as a type 3 system will track a parabolic (i.e. acceleration) input with zero steady state error. A system of type 2 will track a parabolic with a finite steady state error. The system type is inherently type 1 for the control system just described. Therefore, the compensator was modified to increase the system type by adding integral control action. The resulting compensator was a second order over second order. Although the resulting system was not a type 3, this was the best design possible because of software limitations.

3.4.4 Structural Vibrations

3.4.4.1 Structure Description The basic physical structure was derived from a tetrahedron structure best known as the Charles Stark Draper Laboratory design #1 (CSDL #1) (ACOSS III, 1980). It consists of six truss elements, all of equal length, arranged in a tetrahedral arrangement. As seen in Figure 3.11, six short support legs are used to input loads through the bottom vertices. When the term 'truss' is used in structural engineering, it establishes the assumption that members support only uniaxial loads and are therefore pinned at their supported ends. This truss assumption became dissallowed as the space system became more defined.

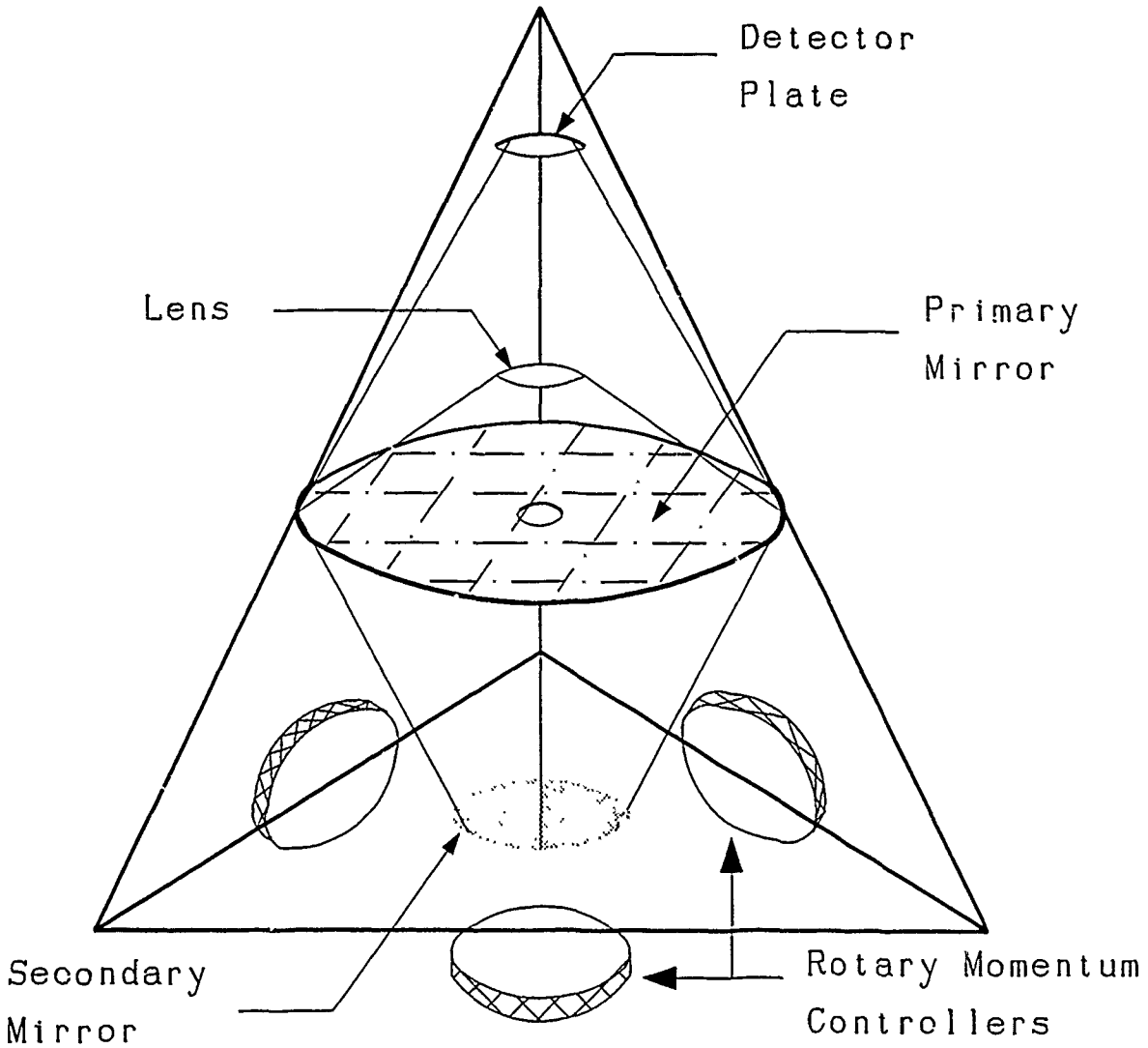


Figure 3.12. System model.

The bare system outline consists of an optical imaging subsystem, a momentum-wheel attitude control subsystem, and a structural subsystem by which the controllers transmit the controlling forces to the optics. The system model is shown in Figure 3.12. In order to gain the maximum rotational interaction between the controllers and the optical subsystem, the connections between the structure and the optical system are considered 'welded.' In other words, the translational and rotational degrees of freedom (DOF) between the structural attachment points and the points of attachment on the optical system are considered as the same. Welded joints imply that bending loads are possible in the structure. With bending loads, a truss structure is no longer possible, and a frame structure is established instead. The frame elements are referred to as beams. The beam end joints attach as welded joints also; this restriction further reduces the DOF of the entire system.

A beam element under a dynamic load exhibits two motions: rigid body and vibrational motion. These motions need to be modeled and input to the optical and control subsystems in an efficient manner during the simulation. The most prevalent manner by which to model this situtation is by finite element analysis (FEA). However, there are several difficulties involved with implementing FEA in the system simulation. Modal analysis is the primary method which is used to perform the vibrations solutions in FEA. Modal analysis is also readily incorporated into bond graphs; bond graphs require far fewer state variables than FEA as well. Bond graphs allow the integration of the structural vibrations model directly into the optical ray trace and controls models. This integration is detailed in section 3.4.3.

The optical and control subsystems have mass and inertia; therefore, they have a direct impact on the motion of the physical system. These subsystems are considered to be much stiffer than the supporting frame subsystem. As a consequence, the optical and control subsystems are modeled as rigid bodies. The individual optical components also have rigid connections between them. This assumption is

established only for the first iteration of the system's design phase. The optical interconnections can be flexible in future iterations. When such an assumption is made, then these additional flexible beams can be immediately incorporated into the modal analysis. The last two major assumptions of the system physical structure are the placement of the optical and controller subsystems relative to the overall frame structure. The optical subsystem is attached at three equidistant points on the rim of the primary mirror. These points are welded to three different beams emanating from the system apex.

These points are never varied; the lengths of these aforementioned beams are five meters. The momentum wheels are fixed at the midspan of the remaining beams. These beams upon which the momentum wheels are fixed will be referred to as 'bottom beams' hereafter. The other three beams are referred to as 'side beams.' A last general observation of the system is that the frame structure is defined by the primary mirror diameter and the side beam lengths. Simple geometry proves that the structure is therefore no longer a completely symmetric tetrahedron, the side beams being longer than the bottom beams.

The above system description hints at the large number of alterables in the physical structure. Since this report is primarily concerned with the simulation of the total system (as opposed to actually manufacturing the satellite), only the beam diameter was varied in the structural subsystem. The other variables were set to nominal values in the simulation.

A detailed description of how the system is modeled in SDRC IDEAS and MSC/NASTRAN is contained in appendix E. The important physical features of the system are summarized below:

1. The optical components do not move relative to each other.
   This feature is called "rigid optics"
2. The optical components are modeled as rigid bodies.
3. The optical system is modeled as a rigid body (due to features 1 and 2).
4. All joints in the system are defined as welded.
5. The momentum wheel controllers are modeled as rigid bodies.
6. The frame structure is composed of flexible beam elements.

These above features are all incorporated into the bond graph model for the structural vibrations of the system.

3.4.4.2 Modal Analysis The vibrational motion of the structure is modeled through modal analysis of finite elements. Any system element has a mass, stiffness, and damping coefficient associated with it. Assuming that these elements are uniform, linear beam elements, there are two end nodes per element which are shared by adjacent elements. The structure is described by a set of differential equations of motion:

$$m\,\ddot{\vec{q}}(t) + c\,\dot{\vec{q}}(t) + k\,\vec{q}(t) = Q(t) \quad . \tag{3.28}$$

where $\vec{q}\ (t)$ is an n-length column vector and n is the number of finite element nodes in the system multiplied by the DOF of the system (Meirovitch, L., 1986). The m, c, and k matrices are n by n in size. These matrices are the mass, damping and stiffness matrices, respectively. The n-length vector, $\vec{Q}\ (t)$, is the forcing vector applied on each DOF for each node in the system. The response motion of the system is a function of both time and position. If the response is assumed as oscillitory, then the homogeneous, undamped equation of motion is best represented as:

$$k\,\vec{u} = \omega^2 m\,\vec{u} \tag{3.29}$$

where ω^2 is a parameter for the eigenvalue problem, and is a natural frequency of the structure. $\overrightarrow{u}$ represents a non-trivial vector derived from the eigenvalue solution. This solution yields a set of diagonal mass and stiffness matrices for the system, called the modal mass and stiffness matrices. Additionally, clever selection of the eigenvectors yield a modal mass matrix with a unity diagonal; this process is called normalization. For n number of elements, there are n solutions for $\overrightarrow{u}$, each $\overrightarrow{u}_i$ corresponding to a particular value of ω_i^2. The eigenvalues, λ_i, which form these solutions, are related to ω_i^2 such that $\lambda_i = 1/\omega_i^2$. The relationships for the normalized modal analysis process are summarized as:

$$\overrightarrow{u}_j^T \, m \, \overrightarrow{u}_j = 1 \tag{3.30}$$

$$\overrightarrow{u}_j^T \, k \, \overrightarrow{u}_j = \omega_j^2 \tag{3.31}$$

$$U = [\overrightarrow{u}_1 \; \overrightarrow{u}_2 \; \cdots \; \overrightarrow{u}_n] \;\; j = 1, 2, \cdots, n \tag{3.32}$$

Equation 3.32 defines a modal matrix which transforms the physical mass, stiffness, and force values into modal space, translating the original set of coupled differential equations into a set of uncoupled equations. This result is due to the diagonal modal m and k. U is also a symmetric and nonsingular matrix; therefore, $U^{-1} = U^T$. An important result of these uncoupled equations is their orthogonality with respect to each other. Combined with the result that ω_i is the natural frequency of the system for corresponding i^{th} mode shape (motion), the separate modes can be evaluated independently. The entire system response to a given $\overrightarrow{Q}$ (t) is then merely a sum of all the modes.

An important assumption in the formulation of the structural model concerns the method used to approximate the structural damping. There are many techniques which accommodate the damping matrix introduced in Equation 3.28. The primary objective of these techniques is to diagonalize c by using U. The model uses an

established modal damping coefficient, such that:

$$\overrightarrow{u}_j^T \, c \, \overrightarrow{u}_j = C_j \tag{3.33}$$

where C_j is the modal damping coefficient for the j^{th} mode.

The modal damping matrix, $[C]$, is by definition a diagonal matrix. Thus, the system motion is decribed completely by uncoupled equations in modal space as:

$$M \ddot{\overrightarrow{\eta}}(t) + C \dot{\overrightarrow{\eta}}(t) + K \overrightarrow{\eta}(t) = \overrightarrow{N}(t) \tag{3.34}$$

where,

$$\begin{aligned} M &= U^T m U = I \\ K &= U^T k U \\ \overrightarrow{N}(t) &= U^T \overrightarrow{Q}(t) \end{aligned}$$

If n is very large, this analysis is very taxing from a numerical point of view. Fortunately, an interesting property of structural modal analysis allows for a reduction in the number of modes actually analyzed. The energies in a given mode become smaller as ω_j increases. As a result, the lowest modes (represented by the lowest vibrational frequencies) contribute nearly all the vibratory motion of the system. These contributing low modes are called the fundamental modes. For physical, three-dimensional structures, fundamental modes are difficult to separate from higher modes, since there is no clear division between frequencies as the eigenvalues increase. In our structural models, slight separations showed between the second and third mode, and then again between the sixth and the seventh mode (see Appendix J). The greater the number of modes modeled, the greater the simulation mimics the modal analysis prediction of vibratory motion. Five modes were modeled in the bond graph for all the structure configurations for the above reasons and also due to a limitation of computer storage space for state variables.

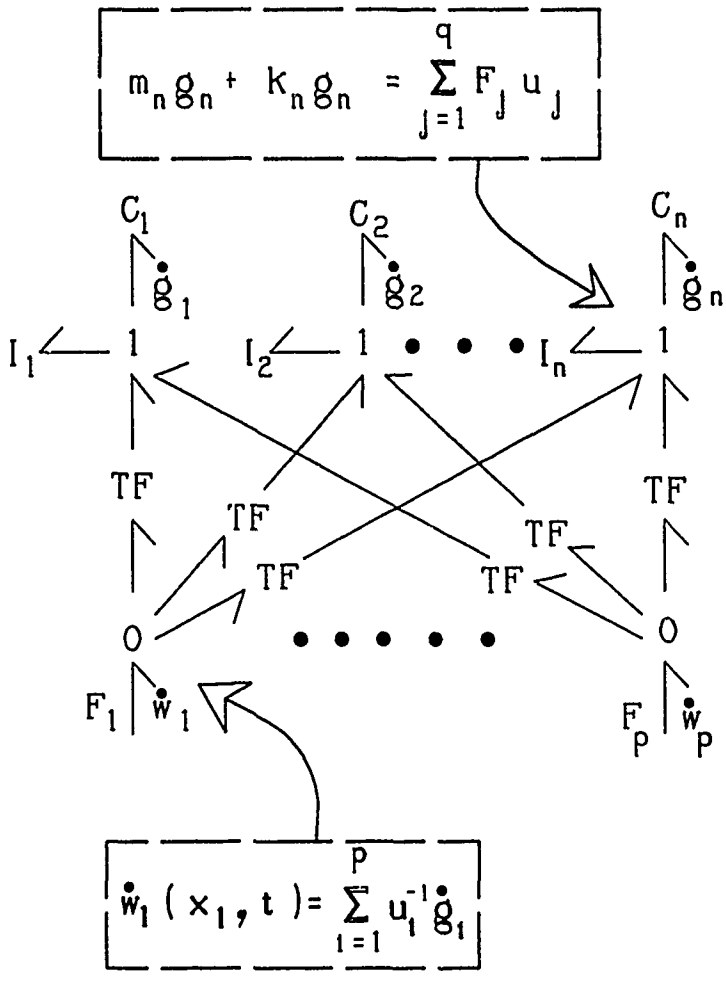


Figure 3.13. Bond graph for a single DOF modal analysis.

3.4.4.3 Bond Graph Modal Representation This research involves implementing a complex system by utilizing bond graph techniques. However, modal analysis has been frequently demonstrated to be feasible for bond graph analysis (Margolis, D. L., 1985) (Margolis and Young, 1977) (Margolis, D. L., 1980).

The uncoupled linear equations to be modeled have already been shown in Equation 3.34. As seen from general bond graph formulation, bond graphs are a graphical representation of state equations. The bond graph represention of Equation 3.34 for a single DOF system with n modes and p force inputs is shown in Figure 3.13. (Margolis, D. L., 1980).

The zero junctions are where the physical forces of the system reactions are summed with the external forces of $Q(t)$. These forces and their respective velocity vectors are transformed by a corresponding eigenvector $\{u\}_i$. The velocity is called a vector due to the nature of having 6 DOF per node location in the full model. The transformers (TF) may be considered a dividing line between the physical displacements and the modal displacements. The compliance, C_i, is an inverse modal stiffness, $1/K_i$. The inductance, I_i, is the modal mass, M_i. In the modal analysis, this value is simply 1. C_i, I_i and $\overrightarrow{u}_i$ are obtained from the eigensolution produced from the NASTRAN solutions. The values for the modal damping matrix C are installed directly into the bond graph for the R element; they are not produced as a result of NASTRAN calculations. The number of bond graph modal 1-junctions will be equal to the number of modes represented and the number of 0-junctions representing physical force summations will be equal to n. The bond graph for the vibrations of the satillite appears as in Figure 3.14.

3.4.5 Optics The optical system consists of a cassegrainian reflecting telescope which collects the optical signal from the source and directs it through a thin lens which focuses it onto a detector. In this study, the telescope-lens-detector system will be referred to as the system sensor. Light propagation through the sensor has been modeled by an exact ray trace program. This program uses as input the missile location with respect to the primary mirror vertex and its vertex normal. The program computes a bundle of 32 rays from the given location to the sensor's primary mirror. The 32 rays strike the mirror at points rotationally symmetric about the optical axis, starting from the inner edge of the center hole in the mirror to the outer edge of the mirror. The main output of this model is the corresponding missile image on the detector. This information is passed on to the system's controllers as a position error between the signal on the detector and the detector's center which is lined up with the optical axis. Other outputs are the mean distance of the bundle centroid striking the detector, relative to the detector center, and the variance of the

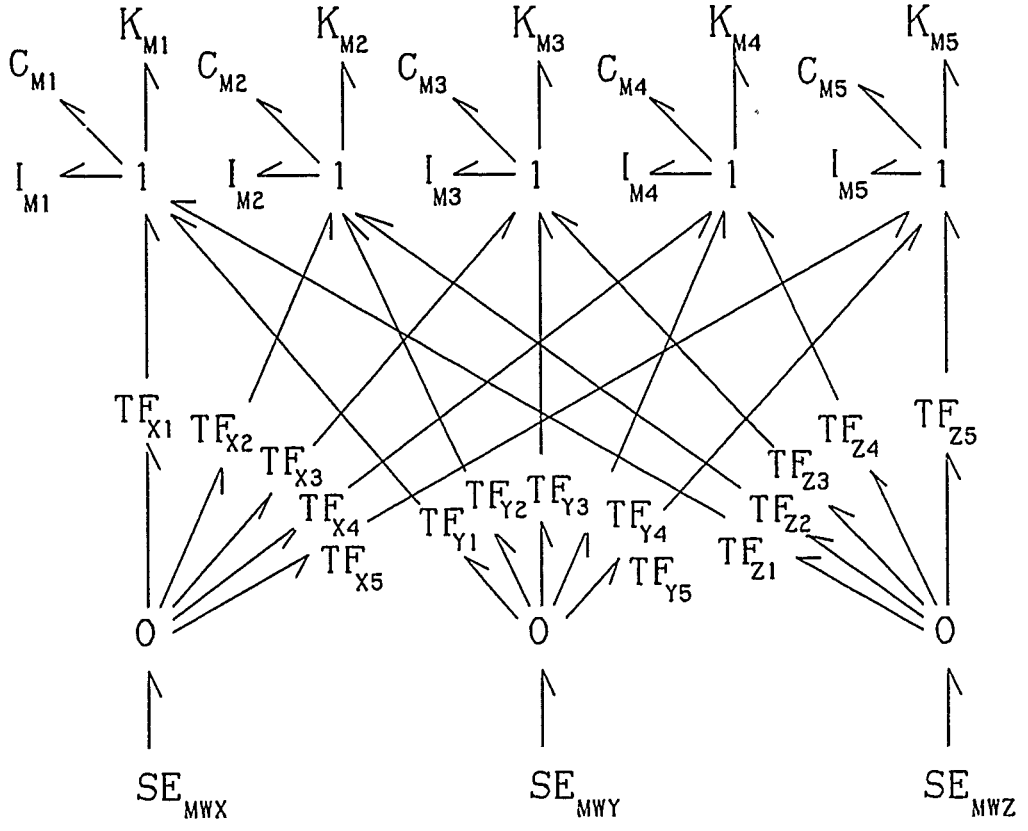


Figure 3.14. Structural vibration bond graph.

rays striking the detector.

The purpose of the following subsections is to present the basic principles of optics which are considered in the design of basic systems such as the cassegrainian telescope. These fundamentals encompass physical optics, geometrical optics, and lens aberrations. These subsections will also include parameter selection of the sensor components, and will conclude with a summary of the exact ray trace representation. These subsections contain more detail than the other model description sections because the intended audience of this research is expected to be more knowledgeable in structural dynamics and controls than in optics.

3.4.5.1 Physical Optics The study of physical optics deals with the wave nature of light. Two of these properties of light are interference and diffraction. Interference occurs when light from a source is divided into two beams which are then superimposed or when any two or more light beams are superimposed. When the beams are superimposed, wave cancellation or amplification occurs. If amplification occurs, it is termed constructive interference, and if cancellation occurs, it is termed destructive interference. This phenomena will not be discussed as it is beyond the scope of this research.

Diffraction, on the other hand, is essential in the understanding of image formation. Diffraction is an inherent optical property which is evident whenever a light wave front strikes an opaque object or when it passes through an aperture. This results in the spreading of the wave front and is known as diffraction (Spindler and Hoyer, 1989).

If the aperture is very large compared with the wave length of light, the spreading of the wave front is minimal and it behaves as a wave front of infinite extent. On the other hand, if the diameter of the aperture is less than half a wave length of light, it will produce a disturbance to emanate as the wave front moves forward, as shown in Figure 3.15.

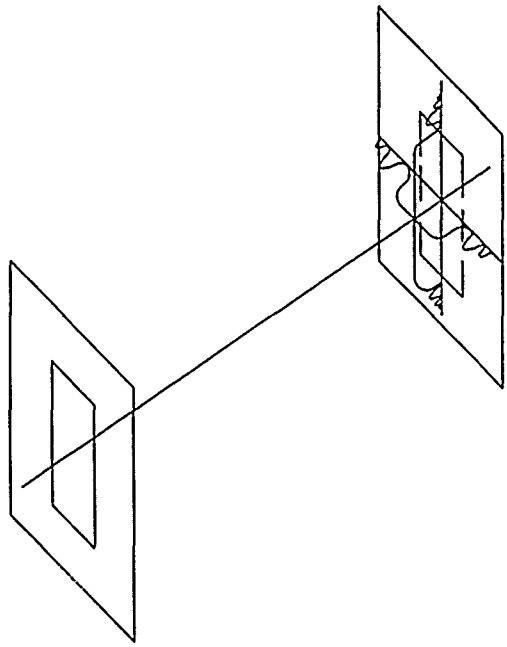


Figure 3.15. Diffraction pattern of a rectangular aperture

Diffraction at a lens is also caused by differences between the lens diameter and the wave length λ. If the lens is infinitely large, the disturbances at every point (except at the focused point) neutralize each other and the illumination is zero except at the focused point; on the other hand, if the lens has a diameter smaller than half the wave length, the wave front illuminates the entire receiving plane. For some intermediate value of the lens diameter, the result is a diffraction pattern as shown in Figure 3.16 (Hardy, A. and Perrin F., 1932).

Since having an infinitely large lens is impossible, the lens will never focus to a finite point. The focused point will look more like a spot. This spot is known as the Airy disk, and its radius is given by:

$$r = \frac{1.22\lambda f}{D} \tag{3.35}$$

where,

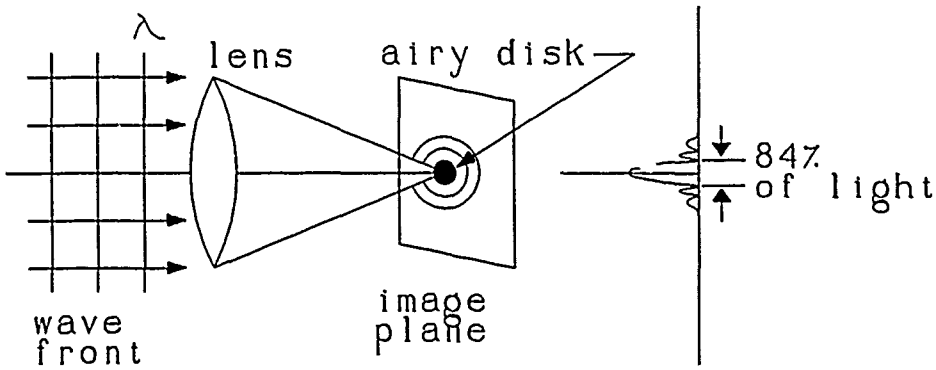


Figure 3.16. Diffraction limited optics

λ = light wavelength

D = diameter of the lens

f = focal length of the lens

Theoretically, the Airy disk contains 83.8 percent of the total energy as shown in Figure 3.16. Diffraction also limits the resolving power of optical imaging systems. Resolving power can generally be defined as the ability of an optical system to separate object points in the image plate. The resolving power is given as the angular separation, θ, at which each point is resolvable (RCA, 1974):

$$\theta = \frac{1.22\lambda}{D} \tag{3.36}$$

3.4.5.2 Geometrical Optics When designing optical instruments, diffraction effects can sometimes be ignored, particularly if all the apertures and optical components are much larger than the wave length of interest. This first order approximation leads into the treatment of light as an ideal phenomena. The branch of optics which neglects the finiteness of the wavelength is known as geometrical optics. Geometrical optics is based on the following fundamental laws.

Law of rectilinear propagation of light Propagation of light through a uniform, homogeneous medium is linear and perpendicular to the direction of oscillation.

Law of reflection When a ray of light is reflected at an interface dividing two uniform medias, the reflected ray remains within the plane of incidence, and the angle of reflection equals the angle of incidence. The plane of incidence includes the incident ray and the normal to the point of incidence (see Figure 3.17). The relationship between the incident and reflected angle is:

$$\theta_i = \theta_r \tag{3.37}$$

Law of refraction When a ray of light is refracted at an interface dividing two uniform media, the transmitted ray remains in the plane of incidence and the sine of the angle of refraction is directly proportional to the sine of the angle of incidence (see Figure 3.17). The law of refraction is also known as Snell's law, and is given as:

$$n_i \sin\theta_i = n_t \sin\theta_t \tag{3.38}$$

where,

n_i = incident medium index of refraction
n_t = refracted medium index of refraction
θ_i = incident ray angle
θ_t = refracted ray angle

The above laws enable the optical engineer to describe the shape of a wave front in simple mathematical terms. This is accomplished by considering various rays of light which are perpendicular to the wave front when the medium is isotropic. This bundle of rays originating at a single point is known as a pencil. For all practical purposes, the light used comes from sources of finite area, every point of which emits a pencil. Such a group of pencils is known as a beam. Using the above principles and geometry concepts, the formation of images is possible.

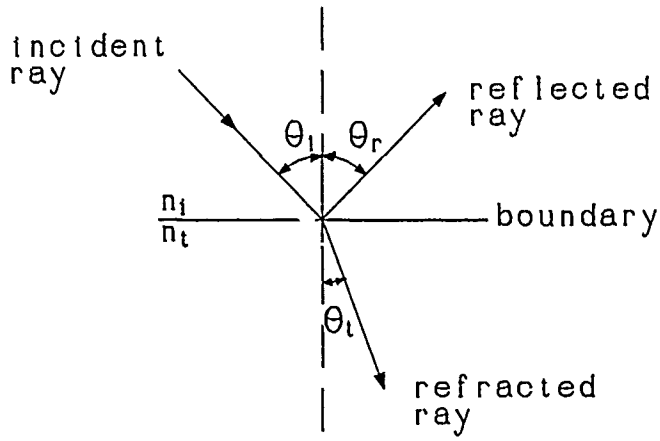


Figure 3.17. Reflection and refraction at a boundary separating two media with refractive indices n_i and n_t.

3.4.5.3 Lens Aberrations Image quality is also affected by optical aberrations. Optical aberrations occur when the rays from an object point do not completely converge at the conjugate image point; furthermore, the properties of the system may be wavelength dependent. Optical aberrations consist of monochromatic and chromatic aberrations but chromatic aberrations will not be considered in this system. Monochromatic aberrations occur with light of a fixed wavelength and have been classified into five types: spherical, coma, astigmatism, image field curvature, and distortion.

Spherical aberration means that the rays from a point object on the axis do not recombine to form a point image. The amount of spherical aberration that is present in a given lens is usually expressed by the distance measured along the axis between the intercept of a ray of interest and the intercept of the paraxial (ideal) ray, as shown in Figure 3.18. Spherical aberration varies approximately as the square of the height of the lens incident ray and depends also upon the distance of the object (Hardy, A. and Perrin F., 1932). Spherical aberration can be eliminated by choosing the reflecting or refracting surfaces in the proper form.

Coma is probably the most difficult to visualize, partly because it can be represented in such a variety of ways due to its nonsymmetry about the optical

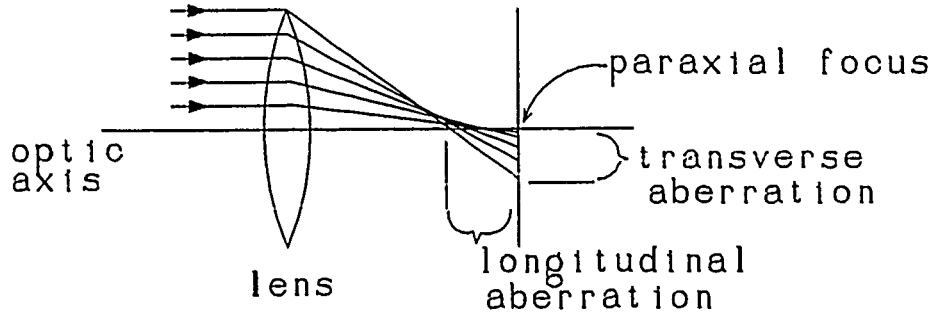


Figure 3.18. Longitudinal spherical aberration in a positive lens.

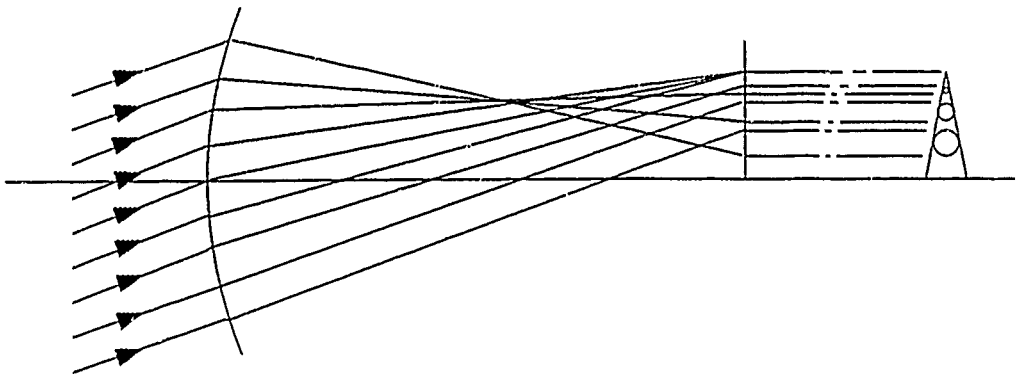


Figure 3.19. Transverse coma

axis. The focal length along the various rays may be different despite the absence of spherical aberration. And since the lateral magnification depends upon the focal length, the magnification will be different for the different zones of the lens. This is not critical for an object point on the axis, but it becomes important as the object point begins to move away from the axis. An additional characteristic is that the zones of the image plate, which are conjugate to those of the entrance pupil, have a tangential orientation forming a cone of 60 degrees. Coma is illustrated by Figure 3.19.

A bundle of rays which fail to combine at a single image point after refraction is said to be astigmatic, and the optical component is said to have astigmatism. Astigmatism, unlike spherical aberration and coma, includes skew rays contained in two planes: meridional and sagittal. Astigmatism increases dramatically as the

object point moves far away from the optical axis. Figure 3.20 illustrates the effects of astigmatism and shows the meridional and sagittal planes. The meridional plane is defined by the optical axis and the object point, and the sagittal plane is defined by the object point and perpendicular to the meridional plane. A bundle of rays from the object point in the meridional plane and lateral to the optical axis strikes a spherical lens surface at different points through the centers of curvature of the lens. Consequently, the radii do not change as far as refraction is concerned. The sagittal plane, on the other hand, does not contain the centers of curvature of the lens surfaces. Thus, different radius of curvature has to be taken into account. As a result, a circular bundle of rays from an object would be imaged as an elliptical beam in the image plane.

For a circular bundle of rays, a horizontal focal line in the sagittal plane is formed by the meridional rays. This focal line is closer to the lens than the focal line formed by the sagittal rays in the meridional plane. Somewhere between both focal lines the elliptical beam in the image plane converts into a circle. This is known as the circle of least cofusion. An astigmatic image can best be understood by considering the point Q of Figure 3.20. If a plate is placed at Q'_1, the image is a line in the sagittal plane; at Q'_2, the image is a line in the meridional plane. At a point approximately halfway between Q'_1 and Q'_2, both lines are of equal magnitude and both lines form the image. This is the best location to collect the image (Hardy, A. and Perrin F., 1932).

Once astigmatism has been eliminated, point images of extended objects can be created. However, the image surface will be curved and imaged onto a flat detector. A detector would have to be curved if the entire image is to be sharply defined on it, as shown in Figure 3.21. This effect is known as curvature of field. It can be shown that the astigmatism and curvature of field of a thin lens are nearly proportional to the focal length and are almost independent of the shape of the lens; aiso, curvature of field is totally independent of object distance.

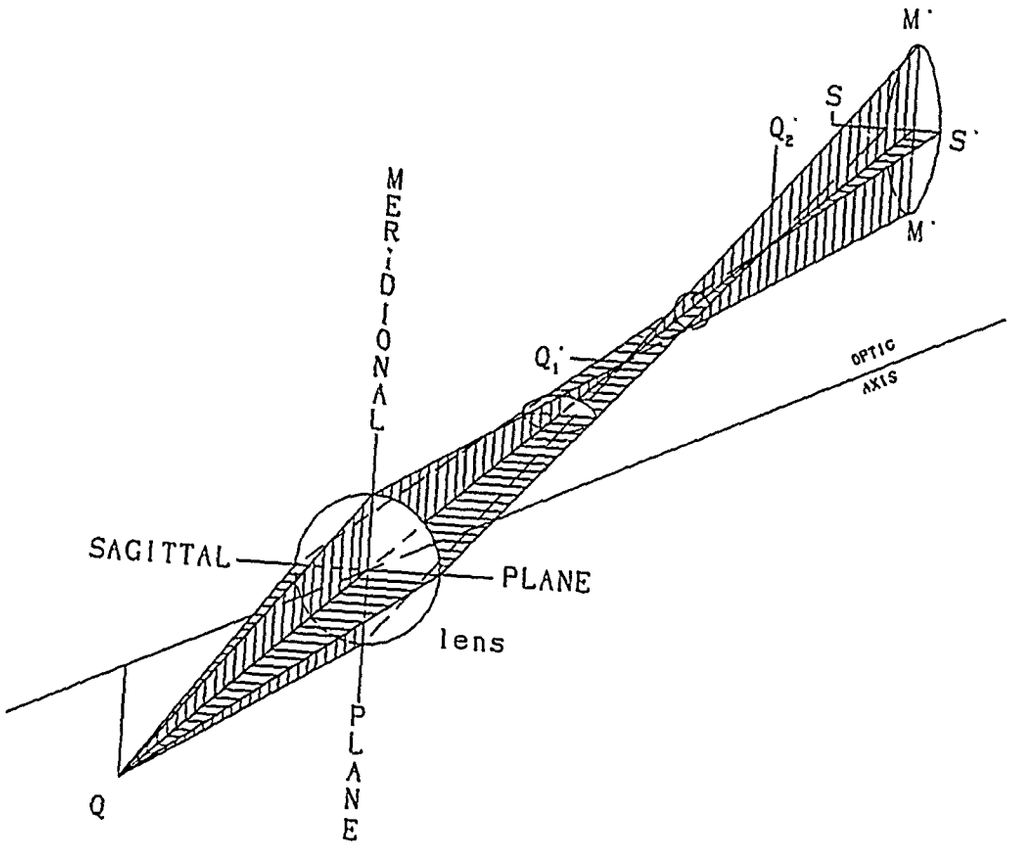


Figure 3.20. Astigmatic images

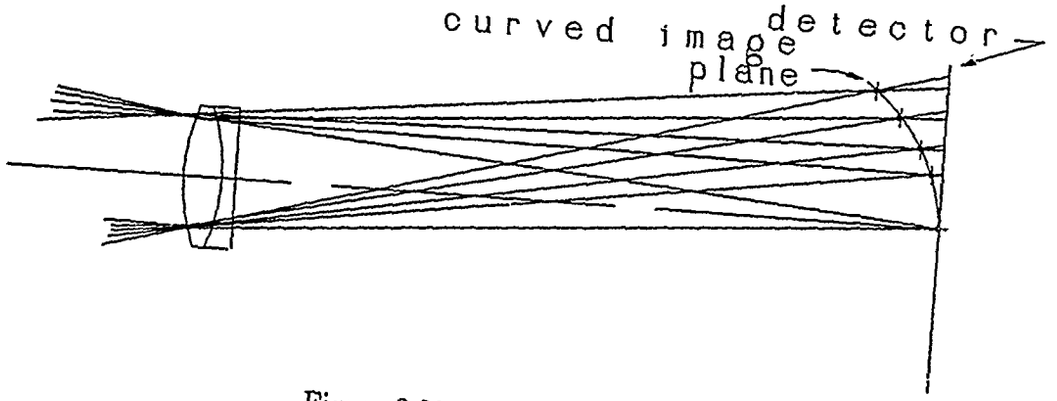


Figure 3.21. Image field curvature

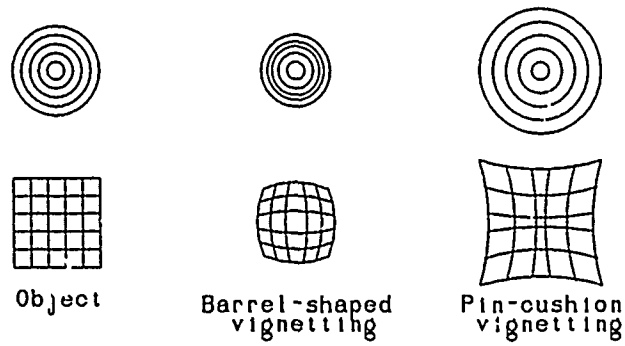


Figure 3.22. Distortion influences on circles and squares

Distortion is caused by variations in the magnification of an object. The magnification varies according to the distance between the object point in question and the optical axis. If the magnification decreases with increasing object size, the diagonals of a square object will be imaged shorter than the actual object. This result is called barrel-shaped distortion. The opposite effect is called pin-cushion distortion. Distortion always occurs rotationally symmetric to the optical axis, as illustrated by Figure 3.22.

The concepts of diffraction, geometrical optics, and monochromatic aberrations were considered in the evaluation of the telescope-lens assembly.

3.4.5.4 Sensor Parameter Selection Since the main emphasis of this research was to study the interaction between the satellite's frame structure, optics, and the control system, the imaging quality of the sensor was not a driving factor for the optical design. The primary concern was to design an optical system with enough mass to affect the frame structure's center of gravity, and at the same time, present a challenge to the control system. As a result of this requirement, a telescope with a large primary mirror was designed. After considering various telescope designs, the reflecting cassegrainian telescope design was selected.

Optical imaging systems are made up of components, which may be classified as objectives, eyepieces and optical relays. The present system consists of a

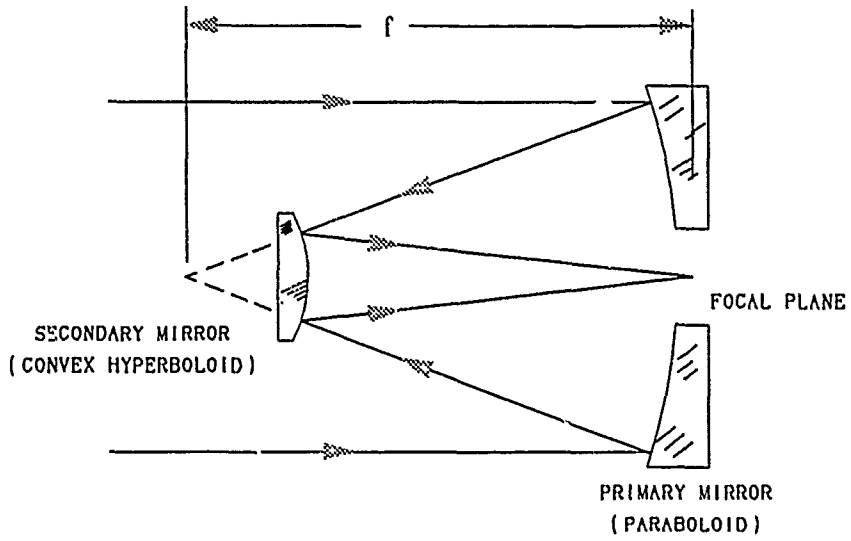


Figure 3.23. Reflecting cassegrainian telescope

cassegrainian telescope, which has two mirrors acting as optical relays; a lens, which also acts as an optical relay; and a detector, which measures and records the image. An optical relay is used in visual instruments to transfer and/or magnify an object by imaging it at a different place and/or a different distance.

A reflecting telescope was chosen because the use of mirrors in place of lenses in optical systems have two advantages: they can be fabricated in much larger sizes than are possible with lenses because they do not need to be transparent, and therefore not completely homogenous. They have no chromatic aberration since the effective index of refraction does not vary with wavelength. Among all the reflective telescopes, the cassegrainian was selected because of its compact design, long effective focal length, wide field of view, and its magnification capability. The cassegrainian design consists of two mirrors: a paraboloid concave primary mirror and a hyperboloid convex secondary mirror, as shown in Figure 3.23. In this arrangement, the primary mirror is drilled with a central hole through which the light is reflected from the secondary mirror. The secondary mirror is interposed in front of the primary focus and a secondary focus is formed at the vertex of the primary mirror. If both mirrors are fabricated to the correct curvature, spherical aberration will not be present.

Once the telescope type had been selected, the mirror diameters and distance between each other had to be determined. The constraint on the maximum diameter of the primary mirror was the size of the cargo bay of the space shuttle, the envisioned launch system. As a consequence, the diameter of the primary mirror was constrained to a maximum of 2.5 meters. Next, the location of the secondary mirror had to be determined. Since the secondary mirror had to be as small as possible to limit interference, it had to be placed very close in front of the primary mirror's focal point. Then the secondary mirror diameter was made big enough to catch all the rays reflected from the primary mirror, assuming that all the rays striking the primary mirror were parallel to the optical axis. Therefore, the primary mirror focal length had to be selected, keeping in mind that it would determine the secondary mirror diameter, and the distance between the primary and secondary mirrors. Along with the focal distance, the image distance had to be determined as well. Equation 3.39 was used to determine the image location of the primary mirror:

$$\frac{1}{d_0} + \frac{1}{d_i} = \frac{1}{f} \tag{3.39}$$

where,

d_0 = object distance

d_i = image distance

f = focal length

To find the image of the primary mirror, an infinite object distance is assumed, and Equation 3.39 gives $d_i = f$, which indicates that the image distance is the same as the focal distance. This assumption is valid only for the primary mirror. Determination of the secondary mirror focal length is next. The object and image distances are already known. The primary mirror image becomes the secondary mirror object. In this case the secondary mirror object is virtual because it is in the back side of the secondary mirror, giving a negative value for the secondary object distance. The secondary mirror image distance is known because it is desired at the

vertex of the primary mirror (see Figure 3.23). Lastly, the focal length is given by Equation 3.39. The next step is to add a thin lens to the cassegrainian telescope design.

A lens was added to the cassegrainian design for three reasons: 1) it provides a second means for focusing, 2) it can be used for magnification, and 3) it is used to change the orientation of the image so that the detector sees it in its proper attitude. The lens diameter and shape must now be considered. The disadvantages of adding a lens to the system are lens aberrations and diffraction effects. Diffraction effects can be reduced by selecting a large lens, but most of the lenses presently being manufactured are small; therefore, the main factor in selecting the lens diameter is in reducing diffraction effects. Since this effect can be reduced by selecting the lens diameter to be greater than half the wave length of interest, a lens diameter greater than 5 micrometers was considered. The next most important consideration for the lens is the shape, because the shape determines the amount of aberration created by the lens. Most of the aberrations could be corrected by inserting a series of lenses of different shapes, but since obtaining a perfect image was not the goal of this design, only one lens was used. The lens was placed as close as possible to the cassegrainian image location, so that a lens as small as possible could be used. The resulting telescope-lens-detector system is shown in Figure 3.24.

The only sensor component left to be selected is the detector. Since this is a conceptual study and the image location on the detector will be given by the exact ray trace program, the only critical parameter in choosing the detector is its dimension. But for the sake of argument, if a real detector was needed, a 3-5 micron focal plane array detector would be used in order to detect the radiation emitted by the missile plume. The dimension of the sensor will determine the sensor's angular field of view if the detector's diameter is the smallest in the system, and normally this is the case. A one-centimeter by one-centimeter detector dimension is very typical for near infrared detection; therefore, this size will be used. The detector was placed

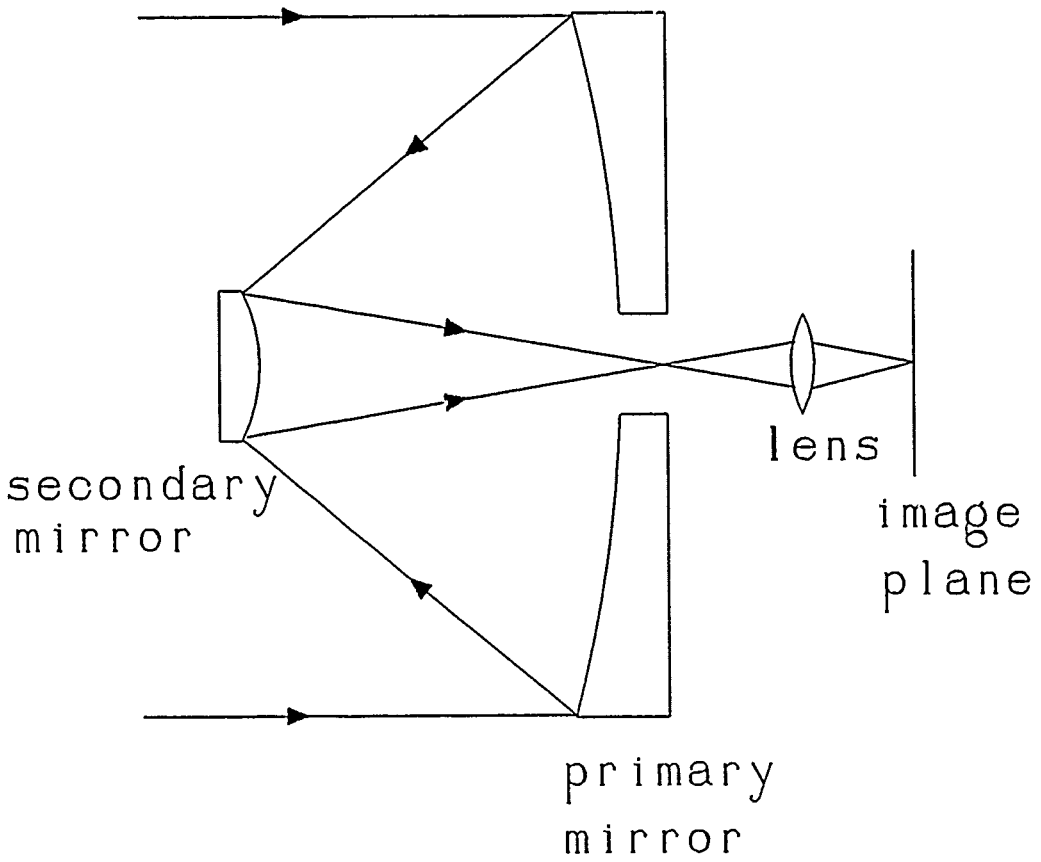


Figure 3.24. Telescope-lens-detector system

Table 3.5. Optical System Parameters

Primary Mirror	Concave
Shape	Parabola
Focal Length	3.5 meters
Diameter	2.0 meters
Eccentricity (shape parameter)	1
Diameter of Hole	0.2 meters
Distance to Secondary Mirror	3.0 meters
Secondary Mirror	Convex
Shape	Hyperbola
Focal Length	0.5 meters
Diameter	0.4 meters
Eccentricity	1.4
Distance to Lens	3.5 meters
Lens	Concave-Convex
First Surface	Aspheric Hyperbola with eccentricity of 1.35
Second Surface	Spherical
Focal Length	0.18 meters
Diameter	0.15 meters
Index of Refraction	2.4324
Distance to Image Plate	0.7209 meters
Image Plate	Flat
Wavelength	3-5 micrometers
Dimensions	1 X 1 centimeter

at the lens image point which is given by Equation 3.39.

The initial theoretical design was evaluated using the exact ray trace program. The goal of the evaluation was to determine the systems ability to transmit the bundle of rays through all the optical components and onto the detector. The initial design had a 2.5 meter diameter primary mirror with a 0.5 meter hole, a 0.8 meter diameter secondary mirror with an eccentricity value of 1.1, a 0.8 diameter plano-convex lens, a 1x1 centimeter detector, and an overall system length of approximately 8 meters.

The first iteration through the ray trace program revealed that only 2 out of the 32 rays made it through to the image plane, but these rays struck outside the

detector. The design was reevaluated as design parameters were changed but this approach proved very time consuming and did not provide a noticeable improvement on image quality. The design parameters that had to be considered were the primary mirror focal length, the secondary mirror focal length and eccentricity, the distance between primary and secondary mirror, the lens focal length, shape and distance from the secondary mirror, and the distance between the lens and the detector. Since there were so many design variables, the system was evaluated component by component. The primary mirror was evaluated first.

This was done by creating a reference surface beyond the primary mirror theoretical focal length so that the intersection of bundle of rays with the optical axis could be found exactly. The coordinates of each strike point on the mirror surface and reference surface were plotted using a plotting software package. The software package provided plots of the bundle of rays from the point source to the primary mirror to the reference surface; consequently, the intersection point of the bundle of rays and the optical axis were found. The theoretical and actual lengths were very close. At this point in the design , it was decided that a smaller primary mirror with a shorter focal length would be used in order to shorten the distance between primary and secondary mirror, and so that a smaller secondary mirror could be used. A smaller secondary mirror creates less interference and increased the light gathering capability of the primary mirror.

The primary mirror diameter was set to 2.0 meters, the secondary mirror diameter was set to 0.4 meters, and the distance between them was set to 3.0 meters. The primary and secondary mirror subsystem, with the new design parameters, were evaluated. This time, there was a big difference between the subsystem theoretical image distance and that found by the exact ray trace program. A difference was expected because theoretical calculations assume spherical surfaces, and neither of the two mirrors is spherical. Using the exact ray trace program and the plotting routine, the desired focal lengths of the mirrors were obtained. A magnified view of

this subsystem focussed point revealed some aberration, probably coma. This effect was minimized by adjusting the secondary mirror eccentricity to a value of 1.4.

The subsystem was evaluated, once again. For this iteration a point source was varied along a vertical line perpendicular to the optical axis, and the subsequent variation in the image space was used to determine the lens diameter. The lens diameter was set to 0.15 meters and the distance to the secondary mirror was set to 3.5 meters. Once again, the mirror-lens combination was evaluated to find the image point for a given object point of interest. The location of the image point became the location of the detector. This step by step analysis yielded the desired design parameters for the exception of the lens shape.

The system with the plano-convex lens had a focussed image point variance of 1E-07 meters. In order to reduce the variance, different lens shapes had to be considered. The other lens considered were two concave-convex lenses, each with a different radii combination, and an aspheric concave-convex lens. The system was evaluated three more times, each with a different lens, and the system with the concave-convex lens yielded the smallest variance, 1E-12; therefore, the concave-convex lens was chosen as the desired shape. All the final values are listed in Table 3.5.

Now that these values had been obtained, the angular field-of-view could be determined. In order to determine the field of view, the system aperture stop and field stop had to be calculated. To find the aperture stop, the element in the sensor whose image in object space subtends the smallest angle with a point source on the optical axis must be found; and then the field stop is the element whose image in object space subtends the smallest angle with the center of the entrance pupil, which is the image of the aperture stop in object space. Figure 3.25 was used as a representation of the sensor for the calculations of the aperture stop.

The aperture stop is calculated using Equation 3.4.5.4 and the values from Table 3.5 as follows:

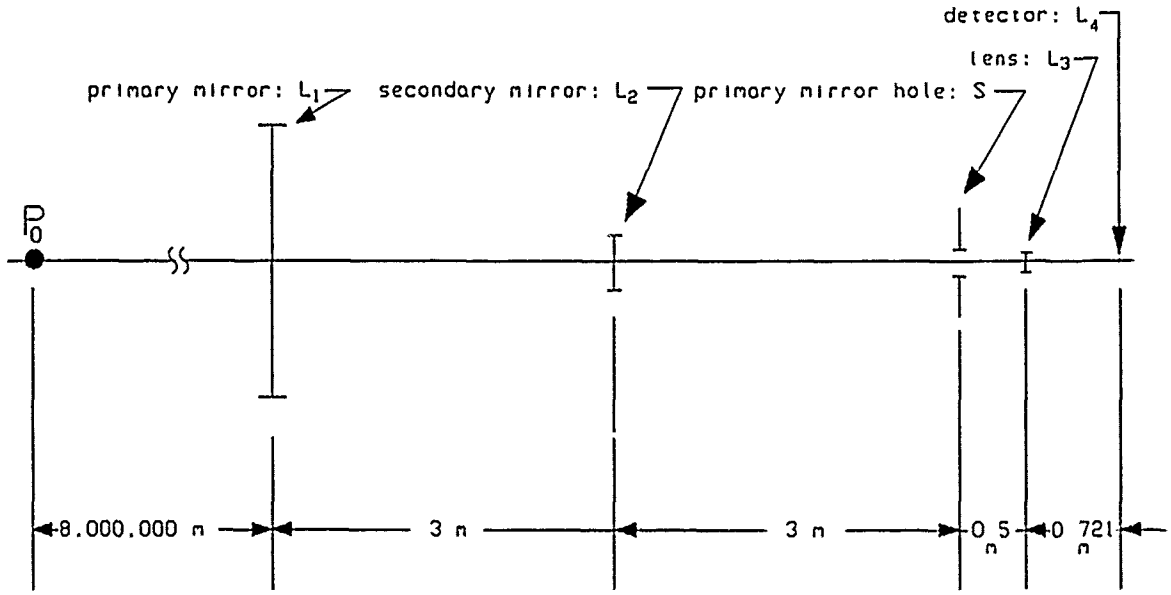


Figure 3.25. Representation of imaging system

1. Half angle subtended by L_1 at P_0 :

$$\phi_{L_1} = \tan^{-1} \frac{1}{8{,}000{,}000} = 7.162E-06 \text{ degrees}$$

2. Half angle subtended by L_2 at P_0 :

$$\frac{1}{3} + \frac{1}{d_i} = \frac{1}{3.5} \longrightarrow d_i = -21.00$$

3. The image diameter (height) is given by

$$y = \frac{d_i}{d_0} = \frac{-21.00}{3.0} = -7.00$$
$$\phi_{L_2} = \tan^{-1} \frac{3.50}{8{,}000{,}021.00} = 25.067E-06 \text{ degrees}$$

4. Half angle subtended by S at P_0 :

$$\frac{1}{3} + \frac{1}{d_{i_2}} = \frac{1}{-0.5} \longrightarrow d_{i_2} = -0.42857$$
$$\frac{1}{3.42857} + \frac{1}{d_{i_1}} = \frac{1}{3.5} \longrightarrow d_{i_1} = -167.99657$$
$$y_2 = \frac{-167.99657}{3.42857} = -48.99902$$
$$\phi_s = \tan^{-1} \frac{24.49951}{8{,}000{,}048.99902} = 175.464E-06 \text{ degrees}$$

5. Half angle subtended by L_3 at P_0 :

$$\frac{1}{3.5} + \frac{1}{d_{i_2}} = \frac{1}{-0.50} \longrightarrow d_{i_2} = -0.4375$$
$$\frac{1}{3.4375} + \frac{1}{d_{i_1}} = \frac{1}{3.5} \longrightarrow d_{i_1} = -192.50$$
$$y_3 = \frac{-192.50}{3.4375} = -56.00$$
$$\phi_{L_3} = \tan^{-1} \frac{28}{8{,}000{,}192.5} = 200.53E - 06 \text{ degrees}$$

6. Half angle subtended by L_4 at P_0 :

$$\frac{1}{0.7209} + \frac{1}{d_{i_3}} = \frac{1}{0.18} \longrightarrow d_{i_3} = .23990$$
$$\frac{1}{3.2601} + \frac{1}{d_{i_2}} = \frac{1}{-0.5} \longrightarrow d_{i_2} = -180.738$$
$$y_4 = \frac{-180.738}{3.43351} = -52.63949$$
$$\phi_{L_4} = \tan^{-1} \frac{26.3197}{8{,}000{,}180.738} = 188.497E - 06 \text{ degrees}$$

Since ϕ_{L_1} is the smallest half angle subtended at P_0, L_1 is the system's aperture stop. The field stop is calculated as follows:

1. Half angle subtended by L_1 at center of aperture stop:

$$\theta_{L_1} = 90.0000 \text{ degrees}$$

2. Half angle subtended by L_2' at center of aperture stop:

$$\theta_{L_2'} = \tan^{-1} \frac{3.5}{21.00} = 9.4600 \text{ degrees}$$

3. Half angle subtended by S' at center of aperture stop:

$$\theta_{S'} = \tan^{-1} \frac{24.49951}{167.99657} = 8.29715 \text{ degrees}$$

4. Half angle subtended by L_3' at center of aperture stop:

$$\theta_{L_3'} = \tan^{-1} \frac{28}{192.50} = 8.27590 \text{ degrees}$$

5. Half angle subtended by L_4' at center of aperture stop:

$$\theta_{L_4'} = \tan^{-1} \frac{26.3197}{180.738} = 8.28537 \text{ degrees}$$

Since the smallest half angle subtended at the center of the aperture stop is $\theta_{L_3'}$, the lens is the field stop; therefore, the angular field of view is:

$$\text{FOV} = 2\theta_{L_4'} = 16.5518 \text{ degrees}$$

This concludes the physical design of the optical system; a description of the exact ray trace methodology follows.

3.4.5.5 Exact Ray Trace Methodology Typical optics systems involve complex geometrical configurations of geometrical shapes used to reflect or refract light rays. The problem with these systems lies in the fact that exact ray tracing can be very difficult for relatively simple systems. As such, a paraxial approach is typically used as an approximation of the actual performance. However, if a method can be found to simply implement an exact ray trace, a more accurate picture of the systems performance can be achieved. This section demonstrates how an exact ray trace method is implemented in the cassegrainian telescope optics system.

The basic premise behind the exact ray trace method is the translation of a given ray from one coordinate system (primary mirror) to another coordinate system (secondary mirror). To begin the ray trace, the orientation of the point in question with respect to the first coordinate system must be known. For the tracking system, all of this data is known beforehand as the exact position of the missile with the satellite is known at all times. For convenience, begin the ray trace inside the optical system and trace from the primary mirror to the secondary mirror. This method can be used anywhere in the optics system without loss of generality. Starting at the primary and tracing to the secondary is done only for convenience.

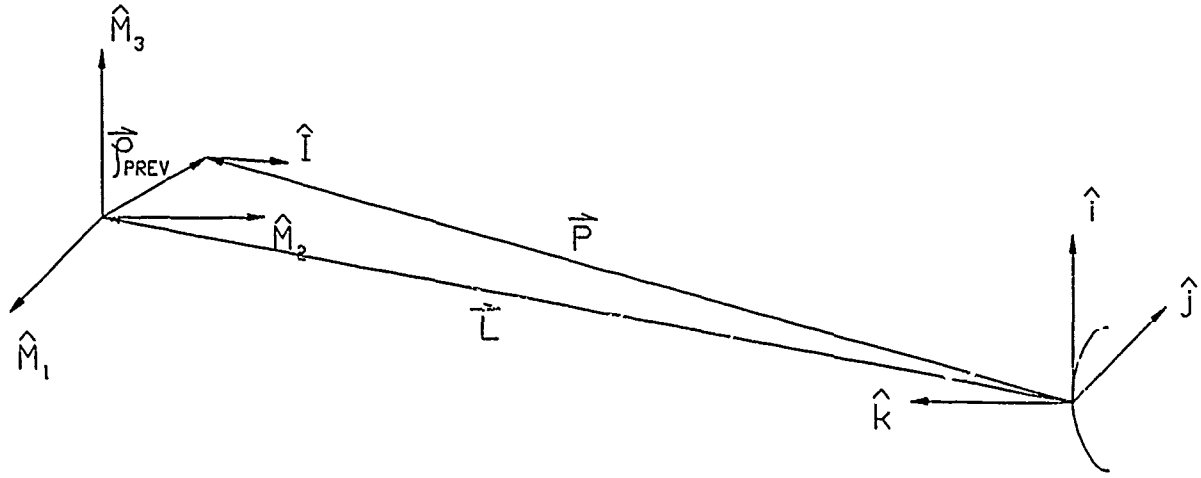


Figure 3.26. Exact ray trace vector definitions

The first step is to attach a reference frame to the secondary mirror. The origin of the frame is attached to the vertex of the mirror with the positive z-axis pointing towards the primary mirror. The x and y axes are oriented tangent to the mirror surface and form a right handed coordinate system as shown in Figure 3.26. The incident ray to the secondary mirror is defined as having a given orientation to the reference frame attached to the secondary mirror. The vector $\vec{\rho}_{prev}$ defines the location of the rays originating point on the primary mirror with respect to the primary mirror reference frame.

Now define a vector $\vec{L}$ that locates the origin of the primary mirror reference frame. The rays originating location can now be defined as

$$\vec{L} + \vec{\rho}_{prev} = \vec{P}. \tag{3.40}$$

Note that coordinate transformations were used between reference frames.

In order to determine the striking point on the secondary mirror surface, it is convenient to first determine the striking point on the x-y plane and then determine the striking point on the mirror from the plane. Defining the distance from the rays originating point to the x-y plane L_1, and define the unit vector in the direction of

the incident ray as shown in Figure 3.27:

$$\hat{I} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{3.41}$$

then the addition of $\vec{P}$ and $L_1\hat{I}$ strikes the plane when the z component goes to zero. Defining

$$\vec{P} = \begin{bmatrix} d \\ e \\ f \end{bmatrix} \tag{3.42}$$

then,

$$L_1 c + f = 0 \tag{3.43}$$

or,

$$L_1 = \frac{-f}{c} \tag{3.44}$$

The trick now is to determine the distance from the incident point on the x-y plane to the mirror surface. For a spherical mirror, the surface is defined by vector

$$\vec{\rho} = \begin{bmatrix} x \\ y \\ z - r \end{bmatrix} \tag{3.45}$$

where,

$$x^2 + y^2 + z^2 = r^2 \tag{3.46}$$

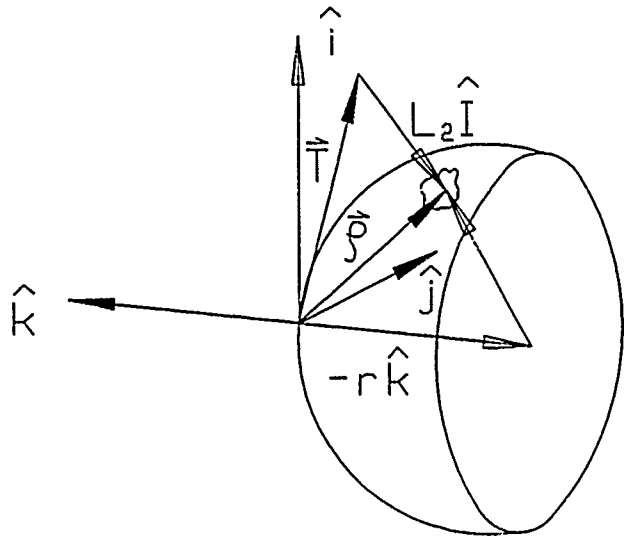


Figure 3.27. Vector definitions for surface striking point.

Now calling the vector from the frame origin to the incident point on the x-y plane, $\vec{T}$,

$$\vec{T} = \begin{bmatrix} g \\ h \\ 0 \end{bmatrix} \tag{3.47}$$

and calling the distance from the plane to the striking point on the mirror L_2, results in:

$$\vec{T} + L_2\hat{I} = \vec{\rho} \tag{3.48}$$

Multiplying this out gives:

$$(L_2a + g)\hat{i} + (L_2b + h)\hat{j} + (L_2c + r)\hat{k} = x\hat{i} + y\hat{j} + z\hat{k} \tag{3.49}$$

where the radius of the sphere has been moved to the left side of the equation. In order for these vectors to be equal and satisfy the solution to the mirror surface

equation, 3.46, the following must hold:

$$(L_2a+g)^2+(L_2b+h)^2+(L_2c+r)^2=r^2 \tag{3.50}$$

Using this expression, the value of L_2 can be determined, and thus $\vec{\rho}$. The reflection off of the mirror is easily defined in matrix notation as:

$$\hat{R}=\left(\mathrm{I}-2\hat{N}\hat{N}^T\right)\hat{I} \tag{3.51}$$

where $\hat{N}$ is defined as the surface normal at the incident point on the mirror surface and $\hat{R}$ is the reflected ray. Once accomplished, $\vec{\rho}$ becomes $\vec{\rho}_{prev}$ for the next mirror, and $\hat{R}$ becomes $\hat{I}$.

This same approach can be extended to a matrix notation (Redding, D., 1989). The general matrix approach centers around one central matrix, the mirror matrix. This matrix is defined as:

$$\mathrm{M}\equiv\left(\mathrm{I}-e^2\hat{\psi}\hat{\psi}^T\right) \tag{3.52}$$

where e is the eccentricity of the geometrical shape of the mirror and $\hat{\psi}$ is a unit vector pointing from the vertex of the mirror towards the focal point. The mirror matrix can be compared to Equation 3.46 as it defines the surface of the mirror, at least in part. In addition, this matrix allows for the principle axis of the conic shape to not lie on any of the principle axes of the coordinate system. This is explained further when flexible optics is discussed. With this definition, the surface of the mirror is defined by:

$$\vec{\rho}^T\mathrm{M}\vec{\rho}+2\vec{N}_o^T\vec{\rho}=0 \tag{3.53}$$

where $\vec{N}_o$ is a vector along the surface normal and magnitude equal to the radius of curvature evaluated at the vertex. The vector $\vec{N}$ at an arbitrary point on the mirror

surface is then:

$$\vec{N} = \vec{N}_o + \mathbf{M}\vec{\rho} \tag{3.54}$$

With these definitions and using the same vectors in the geometric derivation, the following solution can be achieved:

$$\left(\hat{I}^T\mathbf{M}\hat{I}\right)L^2 + 2\hat{I}^T\left(\mathbf{M}\hat{P} + \vec{N}_o\right)L + \hat{P}^T\left(\mathbf{M}\hat{P} + 2\vec{N}_o\right) = 0 \tag{3.55}$$

where L is a scaler equal to the total distance travelled by the incident ray from the previous surface. The reflected ray can then be calculated as was done in the geometric derivation.

The position of the missile with respect to the primary mirror is determined by the orientation of the system in inertial space and the position of the missile in inertial space. This vector acts as the incident ray, $\hat{I}$, to begin the ray trace. The system position is determined by the rigid body rotation model.

3.4.6 Rigid Body Rotation The rigid body rotation model calculates the angles that describe the orientation of the system in inertial space. This model will find the angular rates of the rigid body about the center of mass due to the torques applied by the system controllers. After the angular rates have been found, the angular positions are obtained by integration according to Appendix C.

The rigid body model is based upon Euler's equations. It was developed under the assumption that the torques applied in the model are aligned with the rigid body axes. Euler's equations (Likins, P., '973:page 439) are:

$$M_1 = I_1\dot{\omega}_1 - \omega_2\omega_3(I_2 - I_3) \tag{3.56}$$

$$M_2 = I_2\dot{\omega}_2 - \omega_3\omega_1(I_3 - I_1) \tag{3.57}$$

$$M_3 = I_3\dot{\omega}_3 - \omega_1\omega_2(I_1 - I_2) \tag{3.58}$$

where,

ω_i = the angular rate about principle axis i

I_i = the moment of inertia about principle axis i

M_i = the moment applied to the rigid body about principle axis i

Rearranging slightly, the equations can be written as:

$$I_1\dot{\omega}_1 = M_1 + I_2\omega_2\omega_3 - I_3\omega_3\omega_2 \tag{3.59}$$

$$I_2\dot{\omega}_2 = M_2 + I_3\omega_3\omega_1 - I_1\omega_1\omega_3 \tag{3.60}$$

$$I_3\dot{\omega}_3 = M_3 + I_1\omega_1\omega_2 - I_2\omega_2\omega_1 \tag{3.61}$$

If the body is rigid, then the term $I_i\omega_i$ can be written as the angular momentum H_i of the body about principle axis i. The rigid body rotation model uses the three angular momentum terms as states as a basis in determining the orientation of the body.

3.4.6.1 Rigid Body Bond Graph Description The bond graph of a rotating rigid body is shown in Figure 3.28.

Inertias JSX, JSY, and JSZ are the rigid body's principle moments of inertia. The sources SEMWX, SEMWY, and SEMWZ are the momentum wheels which apply torques to the moments of inertia. The sources apply varying torques according to the control system signals. The modulated gyrators GY1, GY2, and GY3 are functions of the body's angular rate and are modeled as:

$$GY1 = I_3\omega_3 \tag{3.62}$$

$$GY2 = I_1\omega_1 \tag{3.63}$$

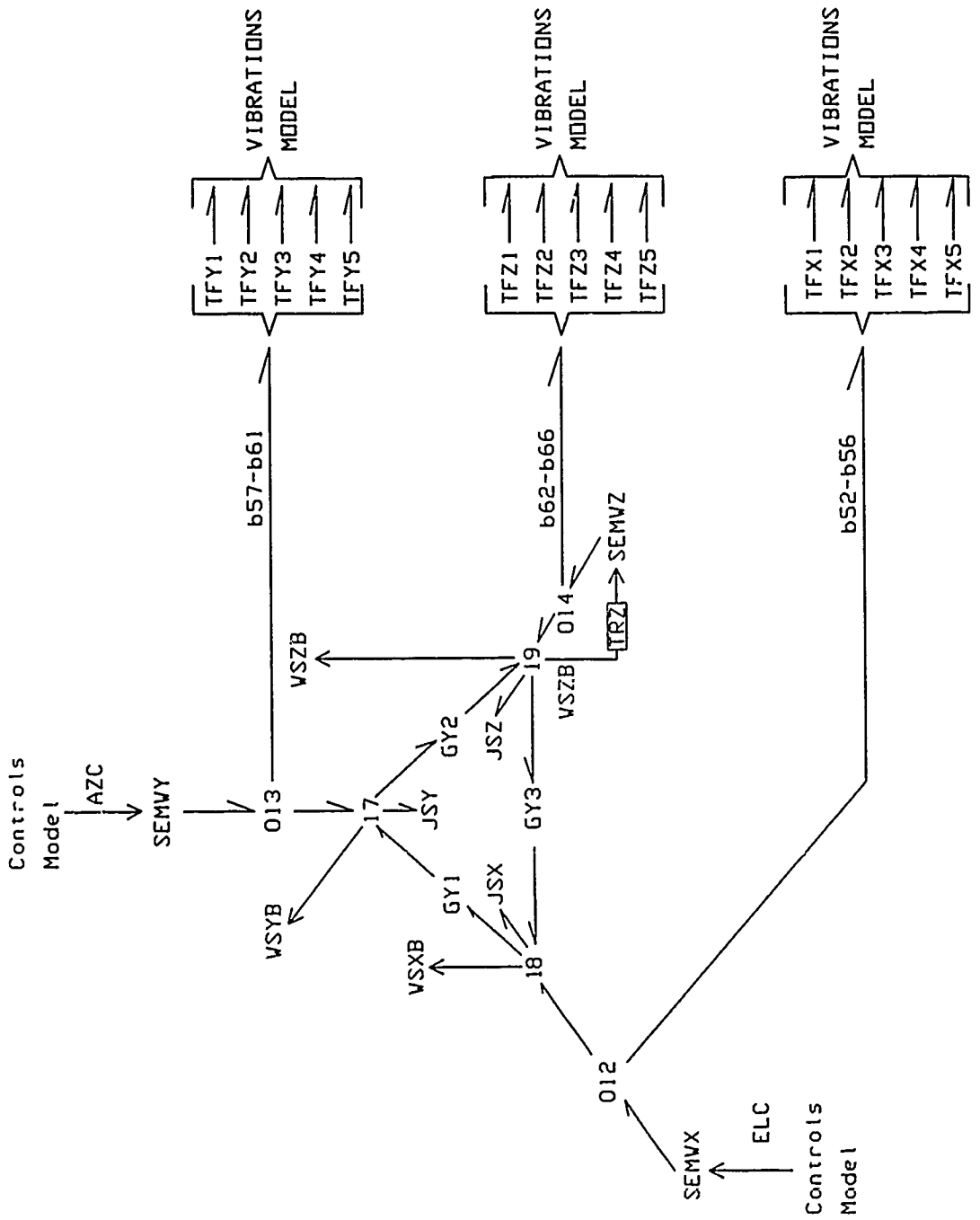


Figure 3.28 Rotating rigid body bond graph

$$GY3 = I_2\omega_2 \tag{3.64}$$

The signals WSXB, WSYB, and WSZB are the angular rates of the body used to determine the angular position.

3.4.7 System Model The design group built and validated six distinct subsystem models. Those models are the missile dynamics model, the system orbital dynamics model, the structural vibrations model, the rigid body rotation model, the optic model, and the controls model. The separate subsystem models were integrated into a single model. Each subsystem model works in a unique manner. The states or variables within each model are not easily combined. A major part of completing the system model is converting the subsystem model outputs into compatible units for interaction with the remainder of the system. First, the general approach of system integration is reviewed.

The first step is to define the units and frames for use in the system model. The frames used in individual subsystem models are: the inertial frame, the rotating Earth frame, the system frame, and the optics frame. The most basic frame is the inertial frame. The inertial frame is based at the Earth's center and does not rotate or translate with respect to the stars. The Earth frame is also based at the Earth's center and rotates with respect to the inertial frame but does not translate. The Earth frame and the inertial frame share a common axis - the Y axis. The Earth frame rotates with respect the inertial frame about the Y axis at the Earth's rotational rate. The system frame is centered at the system center of mass and rotates and translates with respect to both the inertial frame and the Earth frame. The optics frame is fixed at the connection point between the structure and the optics and moves and rotates as the optic system moves with respect to the system frame. The SI system of units is used throughout all the models.

Before integration can be started, the subsystem models must all be in similar states or variables. The individual models employ different types of states or

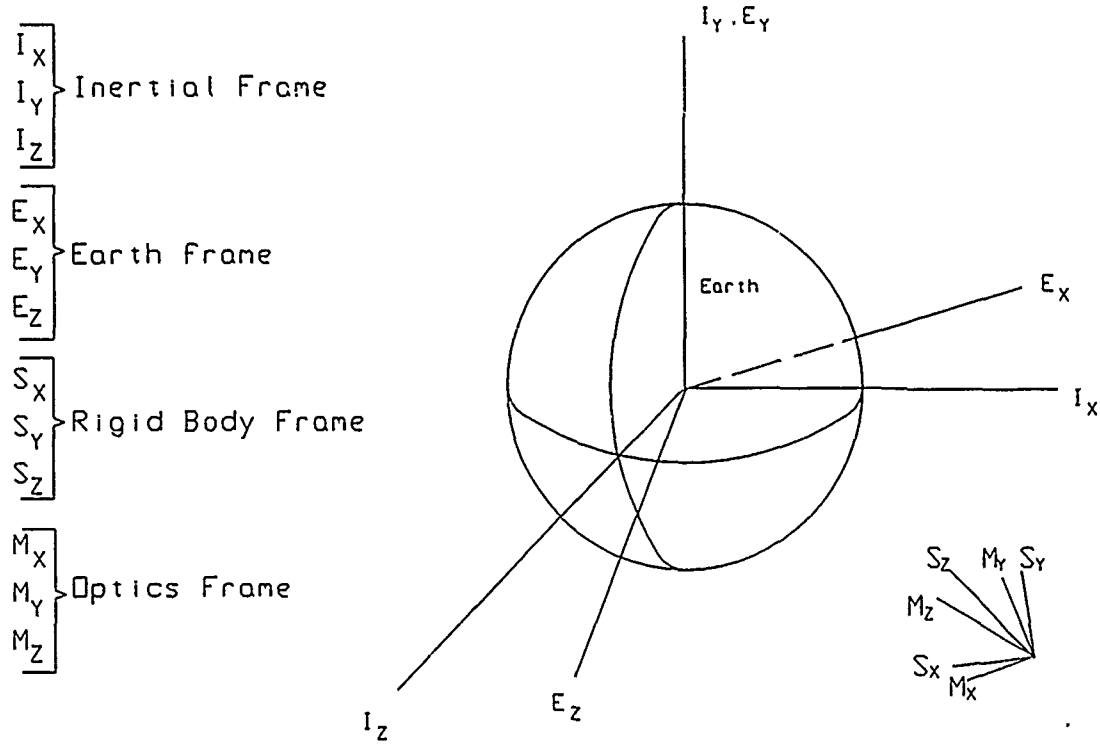


Figure 3.29. System model frames of reference

variables to describe each individual subsystem condition versus time. For example, the missile and system orbital dynamics models integrate accelerations to get linear velocities and the system rigid body model integrates applied torques to get angular rates. The optics model will not accommodate either velocities or angular rates. Therefore, most subsystem models must go through an additional integration to achieve states compatible with the remaining parts of the model. The most commonly used variables are position vectors between subsystems.

3.4.7.1 Missile Dynamics-System Orbital Dynamics Interactions Combined, the missile dynamics-system orbital dynamics models are required to provide the position vector from the system to the missile. A decision was made to express this vector in the inertial frame. The missile dynamics model is expressed in the Earth frame. The system dynamics model is expressed in the inertial frame. The models work similarly. Each model integrates the acceleration to get velocity. The

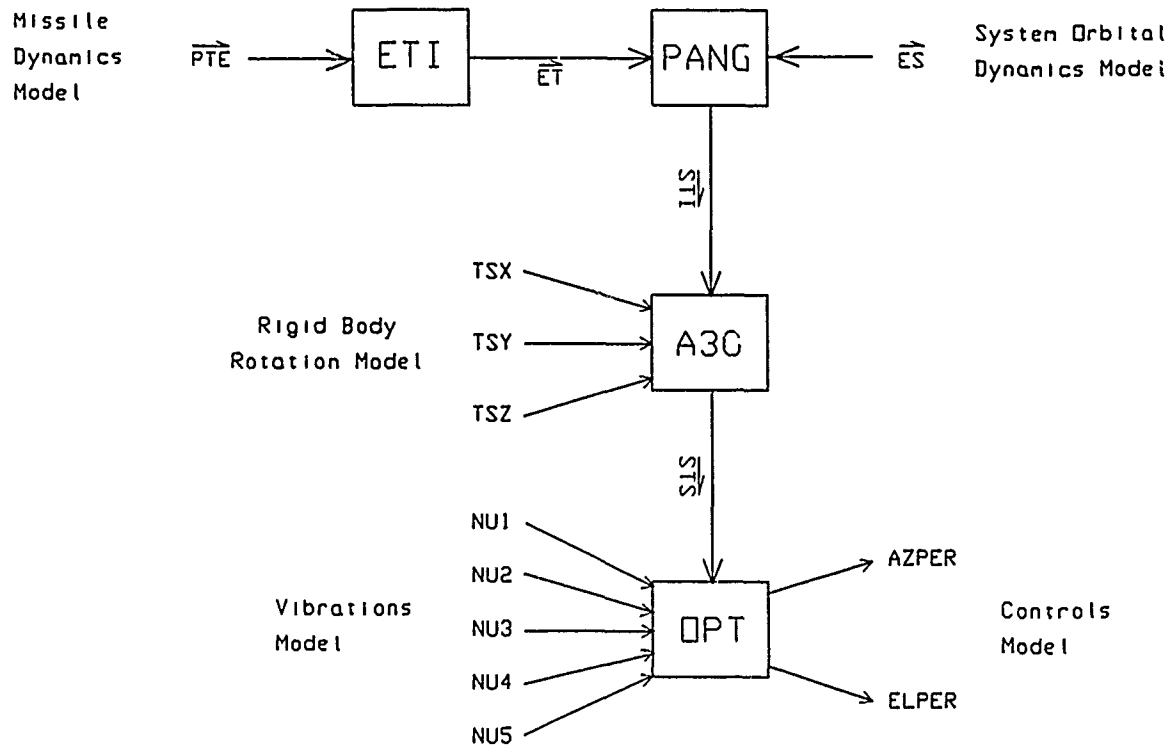


Figure 3.30. Model interactions.

model could difference the velocities between the two systems and then integrate the velocities to get a position vector. The decision was made to integrate each subsystem velocities to get subsystem positions. The positions are expressed in different frames. The missile position is still expressed in the Earth frame and the system position is expressed in the inertial frame. The missile position is then converted from the Earth frame to the inertial frame so that addition to the system position is possible. The function block ETI in Figure 3.30 converts the missile position from Earth frame to the inertial frame. The bond graph signal $\overline{PTE}$ is the position of the missile expressed in the Earth frame. The bond graph signal $\overrightarrow{ET}$ is the position of the missile expressed in the inertial frame. See Appendix C for details of this conversion. Once the position of each subsystem is found in a common frame, finding the position vector from the system to the missile is a trivial matter. The equation providing this vector is:

$$\overrightarrow{STI}=\overrightarrow{ET} - \overrightarrow{ES} \tag{3.65}$$

where,

$\overrightarrow{STI}$ = the vector from the system to the missile

$\overrightarrow{ET}$ = the position of the missile

$\overrightarrow{ES}$ = the position of the system

each expressed in the inertial frame. The function block PANG calculates the vector $\overrightarrow{STI}$.

3.4.7.2 Missile Position-Rigid Body Interactions Once the vector $\overrightarrow{STI}$ is calculated, it is required to express this position in the system frame and then in the optics frame. This section details the calculation of the missile position $\overrightarrow{STS}$ expressed in the system frame. The rigid body rotation model depicts the rotation of a single rigid body in space. The output of this subsystem model is the angular rate of the body in inertial space expressed in the inertial frame. The rigid body rotational model integrates each rotational rate as outlined in Appendix C to get the rigid body angular position. The block A30 then rotates the missile position vector into the rigid body frame using the equation:

$$\overrightarrow{STS}= [{}^{S}A_{I}]\cdot \overrightarrow{STI} \tag{3.66}$$

where,

$\overrightarrow{STS}$ = the missile position vector expressed in the rigid body frame

$[{}^{S}A_{I}]$ = transformation matrix from inertial frame to rigid body frame

3.4.7.3 Rigid Body-Optics Interaction After the vector $\overrightarrow{STS}$ is obtained, the model converts this position to the optics frame. Once this operation is complete, the vector can be input to the optics subsystem model. The position and angular orientation of the mirror is output by the structural vibrations model. This position

and orientation is expressed in the system frame. Using the method outlined in Appendix B, the vector $\overrightarrow{STS}$ is converted to vector $\overrightarrow{STO}$ where $\overrightarrow{STO}$ is the vector from the system to the missile in the optics frame. This operation is performed inside the OPT function and input directly into the optics model. The signals NU1, NU2, NU3, NU4, NU5 are the positions of the 5 structural modes.

3.4.7.4 Rigid Body Missile-Controls Interaction Once the missile vector is found, a method to control the system angular position is required. The only control on the system is the torquing provided by the momentum wheels. The most logical choice of control input is some function of the output of the optical subsystem. The optical subsystem outputs the centroid of the missile image. The inputs of the control subsystem are the angles between the optical axis and the vector $\overrightarrow{STO}$. An easy linear conversion can convert the optical subsystem output into the control subsystem inputs: the azimuth angle error and the elevation angle error. Through testing, it was found that attempting to control the position of the system based upon the output of the vibrating optical subsystem output drove the control system unstable. Several studies have been devoted to the resolution of this problem (Millar, R.A., 1977). However, the solution to this problem is beyond the scope of this research. A further discussion can be found in Section 3.4.3. Therefore, it was decided to bypass the optical subsystem when calculating the input into the controls subsystem. Given the vector $\overrightarrow{STS}$, the following equations provide the two control subsystem inputs:

$$AZPER = \sin^{-1}\left(\frac{STSX}{|\overrightarrow{STS}|}\right) \tag{3.67}$$

$$ELPER = \sin^{-1}\left(\frac{STSY}{|\overrightarrow{STS}|}\right) \tag{3.68}$$

where,

$STSX$ = the x-component of $\overrightarrow{STO}$

$STSY$ = the y-component of $\overrightarrow{STO}$

$STSZ$ = the z-component of $\overrightarrow{STO}$

The OPT function contains the equations performing the above calculations. The signal AZPER is the azimuth angle error and the signal ELPER is the elevation angle error.

3.4.7.5 Controls-Rigid Body Interactions The controls subsystem outputs torque signals to the momentum wheels for pointing purposes. The angles input to the controls subsystem are aligned with angles about the principle axes of the rigid system. The momentum wheels are modeled as being mounted with the axis of rotation on the principle axes also. There is therefore no conversion between angle error control signals and momentum wheel inputs. The momentum wheel with the rotation axis aligned with the system x-axis (X-wheel) controls the rigid body elevation angle error and the momentum wheel with the rotation axis aligned with the system y-axis (Y-wheel) controls the rigid body azimuth angle error. The torque signals output by the control subsystem are supposed to drive the two angle errors to 0. The azimuth angle control is sent directly to the Y-wheel and the elevation angle control is sent directly to the X-wheel. The control subsystem attempts to restrain all rotation about the Z- axis by feeding back the angular rate about the Z-axis through a separate controller. The torque input is then applied to the rigid body system through effort sources. The bond graph signal AZC in Figure 3.28 is the Y-wheel torque control and the bond graph signal ELC is the X-wheel torque control.

3.4.7.6 Controls-Structural Vibrations Interactions When the controls system torques the system to rotate the optical axis towards the missile, vibrations result in the structure. The vibration of the structure will cause the optical system to displace in translation and rotation. The structural vibrations model converts the

control subsystem torques into modal positions and the optical subsystem position and orientation can be determined via the transformation:

$$\overrightarrow{X} = U\ \overrightarrow{\eta} \tag{3.69}$$

where,

$\overrightarrow{\eta}$ = the vector of modal positions

U = the structural modal matrix

$\overrightarrow{X}$ = the vector of generalized positions

Leaving each modal 1 junction (as seen in Figure 3.32) is the signal NUD* (* : number mode) which is the modal velocity. The functions INM* are integrators which determine modal positions (NU*) from modal velocities. Equation 3.69 is performed in function block OPT. Inputs into the structural vibrations model are the modal forces. Modal forces are determined via:

$$\vec{R} = U^T \vec{F} \tag{3.70}$$

where U is defined above, $\vec{R}$ is the vector of modal forces and $\vec{F}$ is the vector of generalized forces. The bonds b52-b56 in Figure 3.28 are the X-wheel torques input into the modal bond graph. The bonds b57-b61 are the Y- wheel torques input into the modal bond graph and the bonds b62- b66 are the Z-wheel torques input into the modal bond graph. The transformers TF(X,Y,Z)(1,2,3,4,5) perform the operation described in Equation 3.69. The modulator of each transformer is determined by the individual entries of U.

3.4.8 Model Validation Each subsystem model was validated both before and after integration into the system model. Without alternative sources of data, the design group had no way to determine the validity of individual models. The approach adopted by the group was to check the model for correctness in basic

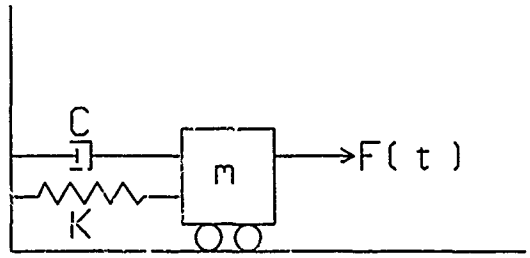


Figure 3.31. Mass-spring-damper system

simulations with predetermined inputs and to compare model outputs to expected results. For validation purposes then, most models were subjected to simple inputs. The system and missile dynamics models were validated using external data sources.

3.4.8.1 ENPORT Validation The first step in the validation process was to ensure the ENPORT modeling software correctly solved for the time response of the model. The design group modeled several simple mechanical systems in ENPORT for validation purposes. The most well known validation system was the simple mass-spring-damper system of Figure 3.31

The mass, spring constant and damping factor were chosen so that $\omega_n = 1$ rad/sec and $\zeta = 0.5$. ENPORT solved the time response of the system for a unit step input. Data was collected and compared to expected results (Reid, 1983:pages 88–99). Other simple models were modeled and run on ENPORT after which the output was compared to expected results. ENPORT was found to model the simple systems without error. All ENPORT integration methods were tested using simple models and were found to correctly integrate the system states. Using the results of the simple model testing, ENPORT was accepted and the group proceeded to validate the subsystem models.

3.4.8.2 Missile Dynamics An external data source was used to validate the missile dynamics model. A SLAM II simulation was constructed to simulate the flight of a missile under thrust. Once this model was determined to be correct, it was

used as a control for comparison to the ENPORT model. Within a sufficient accuracy, the two models produced the same result and the missile model was accepted as validated.

3.4.8.3 System Orbital Dynamics The system dynamics model was very similar to the missile dynamics model except the system dynamics model did not contain any effort sources for thrust. The only force upon the system is gravity. The model was tested by setting the initial conditions (positions and velocities) such that the resulting system was in a circular orbit. The position of a body in a circular orbit is quite easy to calculate. (Greenwood, 1965:pages 200–209). When the system was placed in a geosynchronous orbit with the correct geosynchronous orbit velocity, the ENPORT model solved the model exactly. After this testing, the system dynamics model was accepted.

3.4.8.4 Rigid Body Rotations The rigid body model testing was similar to the ENPORT testing. The rigid body model simulates rotation about the three coupled principle axes. It was possible to simply rotate the body about single axis without interaction with the two remaining axis. An effort (torque) source was independently applied to each axis. ENPORT solved the model correctly. The rigid body model was thereby accepted.

3.4.8.5 Controls After the rigid body rotation body model had been accepted, the next step was to incorporate a control system. This subsystem model was tested several different ways. One method employed was to command the controller to rotate the system about one axis to a particular position.

After validation of the controls model was complete, a subset of the total model was tested together. The above subsystems were combined to model a rigid body orbiting the Earth at geosynchronous orbit. The control system was set to point one principle axis directly at the center of the Earth. During this testing, the

missile position was replaced with the position of the center of the Earth (0,0,0 in the inertial frame). If the correct control algorithm was applied, it was discovered that the mini-model could accurately simulate a rigid body 'tracking' the center of the Earth. This was a large step toward the total model validation. It was felt if the model could track the Earth's center, only a small change would be required to simulate the tracking of the missile. The Earth position would be replaced with the missile position from the missile dynamics model. The final two subsystems to integrate were the structural vibrations model and the optics ray trace model.

3.4.8.6 Optics Ray Trace Model Validation of the optics program was a long and tedious process because of the complex geometry involved in the program. However, the problem was completely deterministic and was handled routinely once the process was defined.

The problem involved determining if each step of the optics calculation performed as intended. This was compounded by the fact that each step involved long and complex calculations, each of which had to operate together in order to obtain results that had meaning. To reduce the problem, each subroutine was checked individually to verify their correctness and completeness. This was accomplished by applying known quantities as inputs and verifying the output reflected the proper response to the given input. Although complete debugging of every subroutine was, in all likelihood, not completely achieved, the subroutines were tested in a variety of ways to account for every possibility we felt the system would encounter. Undoubtedly there are still many situations that were not tested as complete testing of software is never fully realized.

Once all of the subroutines were debugged to a high level of satisfaction, the subroutines were tied together into a complete package. The entire package of subroutines was required in order to complete a ray trace from one component to the next. The operation of the software is included in Appendix F and will not be

repeated here. The reader is referred to Appendix F for the details but a brief description follows. The incident ray of light originates on a component at a known position and orientation. Utilizing this information and the known shape of the current component, the program calculates the tracing of the ray from the first component to the current component and calculates the point on the surface of the current component where the ray strikes. A surface normal at the point of incidence is calculated. Utilizing the direction of the incident ray and the surface normal, the ray is reflected off of the component and thus becomes the incident ray for the next component.

This process was carried out as a whole as most of the calculations performed were used in each step. The techniques utilized in Appendix F were used to hand calculate a ray trace from one component to the next and calculate incident points, normals, and reflected ray directions. Data for each ray was stored in data files and later recovered for accuracy checking. This was a difficult process as each ray required hand calculations and comparisons with the program data. This process was meticulous and required several hours of work for each run. In addition, the runs were made by changing the shape of the components as well as checking incident ray calculations on concave and convex surfaces.

Tables were set up to define what type of component was being used as the software had to configure itself differently for each kind of component. Once accomplished, the software performed correctly. The validation continued for each component as new components were added to the optical system. Once the system was defined and the software finalized, a final check was made by ray tracing using the software and comparing with hand calculations from a given point source to the image plate. This completed the validation process and resulted in a validated optics program model. After the optics model was validated, another subset of the total model was tested for validity. The optics model was integrated into the previously tested model. The largest change from the above model was change of control sys-

tem input. In the previous model, a converted missile position vector in the rigid body frame was the control input. In this model, the missile position vector was run through the optics model. The optics model output was then converted into the controls input. Again, this model worked as expected. The simulation showed an optical system in geosynchronous orbit tracking the center of the Earth. The only model left to integrate into the total system was the structural vibrations model.

3.4.8.7 Structural Vibrations Model In testing the structural vibrations model, it was desired to compare the vibrations model output to expected results. However, the response of this particular structure to a dynamic input was not available for reference. Therefore, the structural vibrations model was validated using a simple structure and a simple input. A one inch diameter, 1 meter length, pinned end beam was modeled in the same modeling technique. A 1000 newton step force was applied at the center of the beam. The model output for four different points along the beam were compared to theoretical results of the same situation (Shigley, 1977:page 642). The structural vibrations model was found to correctly model this simple structure and it was assumed that a more complex structure would also be modeled correctly.

3.4.8.8 Total Model The complete model was assembled after the structural vibrations model was validated. The complete system bond graph can be seen in Figure 3.32. A missile position vector in the inertial frame was calculated based upon output from the system orbital dynamics and missile dynamics model. Using the rotational position output from the rigid body model, the missile position vector was transformed to a new missile position vector in the rigid body frame. Using the structural vibrations model, the same vector is now transformed to the optical frame. This transformed vector was then input into the optics model. The optics model is then fedback into the controls model which commands the momentum wheels. The momentum wheels torques are input into the rigid body position and the structural

vibrations model. The same simulation was attempted, to track the center of the Earth. With a vibrating optical model output as the control model input, the system was found to be unstable. Designing a control system to damp out vibrational modes was outside the scope of this research, so the system was redesigned. The model was changed so that the control system input was the converted missile position vector in the rigid body frame, not in the optical frame. The system was stable in this configuration and the optical model continued to output the image position and variance data. That output was now collected for imaging analysis purposes. Using this configuration, the system easily tracks the center of the Earth as it orbit at geosynchronous altitude. The model was then changed to track the missile. All data collected indicates the model correctly simulates the system as it follows the missile during the 360 second mission time.

3.5 SYSTEM OPTIMIZATION

A methodical approach for arriving at the optimal values for the system design variables was necessary in order to efficiently determine the trends of the system's performance. This effort resolved into the areas of reducing the number of variables, following a methodical plan for collecting performance data, and using empirical modeling techniques to characterize the system. The results were then optimized for the region of the satellites' operational design region.

3.5.1 Reduction of Variables The number of design variables was decreased to reduce the complexity of the problem and to enable the system design to be more clearly characterized. This process reduced the system design variables from 24 to only four. Twenty-three design variables came from the design level alterables of table 3.2; one was a system level alterable: orbit radius. The interaction matrices of Figure 3.1 were referenced to insure coupling of the selected design variables. In other words, the smallest number of variables were to be retained while each still affected at least two subobjectives.

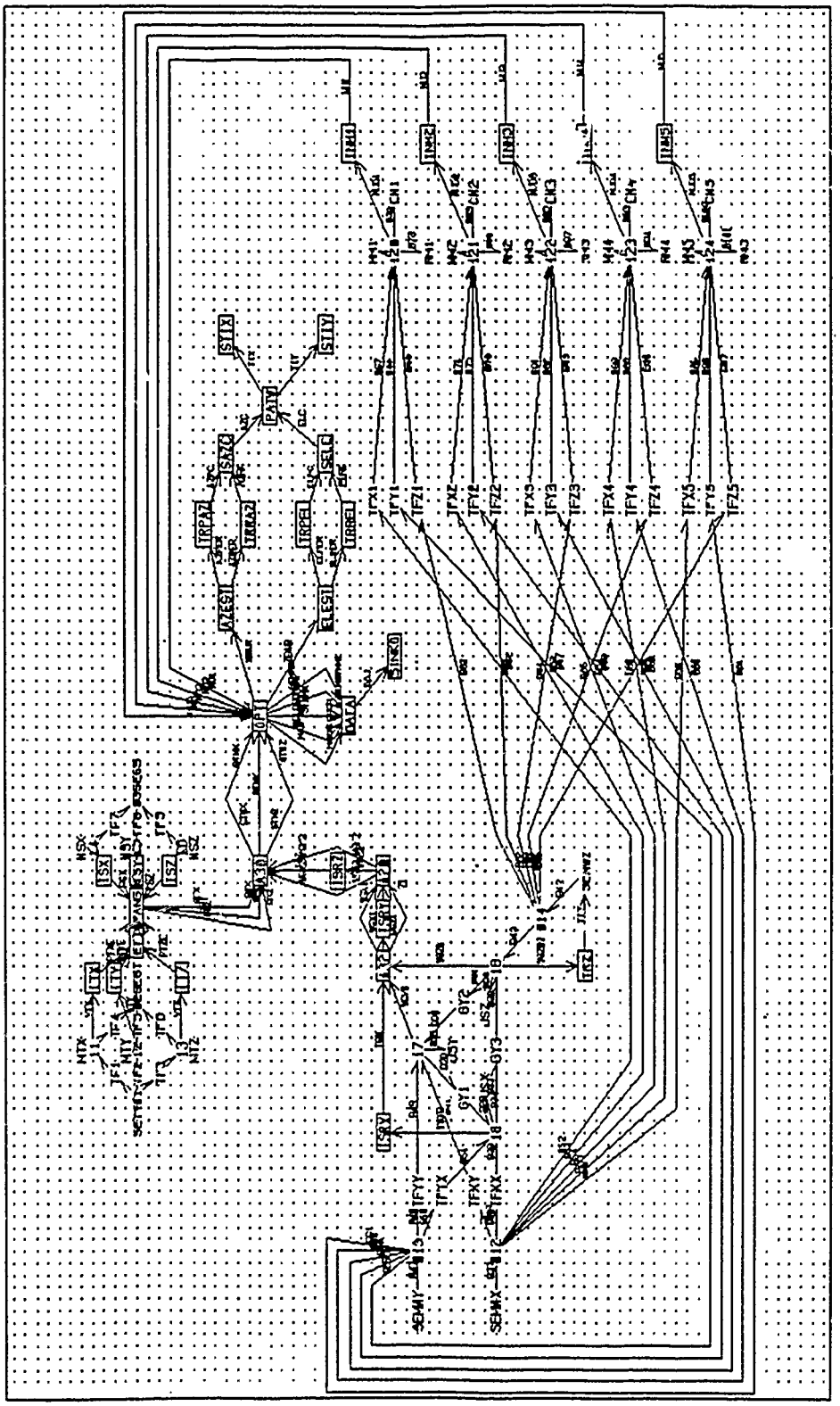


Figure 3.32. Total system bond graph

With the exception of the primary mirror diameter, all of the optical variables are dependent on the orbit of the system. This assertion is based on the theory of optical design detailed in this chapter. Therefore, the optical system can be characterized by the system orbit. The system orbit interacts with the image quality, detector plate power, and tracking error subobjectives.

Four of the five control subsystem variables influence at least three subobjectives. The exception, the momentum wheel masses, are direct functions of the total system mass (Agrawal, B. N., 1986); therefore, these alterables are determined by the entire system configuration. This restriction eliminates the momentum wheel masses as a true design variable. The controling subsystem can be characterized by any of the remaining control variables; settling time was selected as the control design variable and the rest were influenced by its setting.

Only two of the structural framework design variables, the damping coefficient and the controller placements, influenced at least two subobjectives. Modal damping coefficients of any structure are often arguable from one school of engineering to the next. Since the system study is concerned with the development of the satellite and not of space structure vibration research, the coefficient was set to a conservative value. The momentum wheel placement was a problem to characterize. The momentum wheel mass determination required that they be as near the system center of gravity as possible (Agrawal, B. N., 1986). Thus, this alterable was quite restricted and was subsequently eliminated as a design variable. Obstruction of the optical field of view by the momentum wheel controllers will always be present in the conceived system configuration. Varying the side beam lengths would present an additional complication to this problem. Therefore, the side beam lengths were set to a nominal value of five meters. The stiffness of the structure thusly became dependent on the beam diameter and the beam materials used. Since the beam diameter is a continuous variable, it was selected to be the structural framework design variable. The beam material was selected as a nominal value which is derived in Appendix E.

In reviewing the three design variables selected and referencing Figure 3.1, it was noted that the subobjective of Minimize Mass was not coupled to the influences of the other subobjectives, either by direct interaction or through the reduced design variables. Therefore, a fourth design variable was needed. The primary mirror mass was a significant portion of the total mass of the system. This component's mass is characterized by the primary mirror diameter. The interaction matrix of Figure 2.4 shows that this diameter impacts the subobjectives of image quality and detector plate power as well. This strong influence across the subobjectives resulted in the selection of the primary mirror diameter as the fourth design variable.

The four design variables for the satellite are:

1) Controller settling time
2) Orbit diameter
3) Primary mirror diameter
4) Beam diameter

3.5.2 Optimization Approach The following design variables were selected: orbit radius, control system settling time, structure beam diameter, and primary mirror diameter. The measures describing the performance of the system are the optical system mean track error (MTE), inverse of light intensity upon the primary mirror (ILI), light intensity upon the detector divided by the variance (signal to noise measure) (IVR), and system mass (SM). Performance measure responses to changes in design factors are not easily determined. To optimize this system, it is desirable to find the design factor settings that optimize the responses of the performance measures. A plan was necessary to efficiently gather data in order to form an empirical function describing each performance measure. This planning and estimation procedure was based upon Response Surface Methodology (RSM) methods.

3.5.2.1 Empirical Models Given a response y which is an unknown complex function of n design variables, $z_1, z_2, ...z_n$, it is possible to estimate y over a small region of z_i by a linear equation of the form:

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i \tag{3.71}$$

where m is the number of number combinations of the z_i variables and design factors x_i are polynomial combinations of z_i. If $\vec{y}$ is a vector of p observations of y, X is the p by m matrix of design factor settings and $\vec{\theta}$ is the vector of m β_i parameters, then the response of the system can be written as:

$$\vec{y} = X\vec{\theta} + \vec{\epsilon} \tag{3.72}$$

where ϵ is a vector of m random errors. Each element of ϵ is assumed to be independently and identically distributed. The parameters of $\vec{\theta}$ are unknown before the observations are collected. However, using least squares regression, $\vec{\theta}$ can be estimated using the equation:

$$\hat{\theta} = (X^T X)^{-1} X^T \vec{y} \tag{3.73}$$

where $\hat{\theta}$ is the unbiased estimator of $\vec{\theta}$ as long as the columns of X are orthogonal. Given a computer simulation of a system, it is then possible to specify the factor settings for a number of runs to collect the data required to perform the least squares regression of the parameters of Equation 3.71. After $\hat{\theta}$ is obtained, then the unknown response y can be estimated via:

$$\hat{y} = \hat{\beta}_0 + \sum_{i=1}^{m} \hat{\beta}_i x_i \tag{3.74}$$

3.5.2.2 Central Composite Design The Central Composite Design (CCD) is but one of many methods used in RSM to methodically obtain the data required for an empirical mathematical model. This design was selected for this research because a second-order model was desired, and CCD requires very few samples of data while allowing for an orthogonal regression analysis to be performed.

CCD is based on the requirement that a second-order empirical model is to be determined. Each design factor is set at one of five possible settings for a design simulation: $-\alpha$, -1, 0, +1, $+\alpha$. The α values will be discussed later. For different combinations of design values, the responses y are collected.

The number of computer simulations from which to collect data must be as small as possible. This requirement is a direct result of the time consumed for each design's computer simulation: approximately 3 to 5 hours, real time. Searching all possible design factors values to collect data is an impossibility, since these factors are continuous. The three levels of the design factors could be searched for all possible values; this method is called factorial analysis and is statistically very powerful. Unfortunately, a full second-order factorial design requires $N = 3^4$ computer simulations. The CCD design, however, requires the fewest observations (Box and Draper, 1987):

$$N = 2^k + T \tag{3.75}$$

where, N is the number of observations required, k is the number of design factors, and T is the number of center and axial points. The value for $T = 2^k + 1$ assumes only one Y value is collected when all the design factors are set to their central values (the center point). The research case requires $k = 4$, $T = 9$; therefore, the minimum number of observations required is $N = 25$. This quantity is much preferred over the full factorial design requirement of 81.

In order for the design matrix to be completely defined, the axial points, α,

Table 3.6. Levels of the design variables

θ_1	θ_2	θ_3	θ_4
-1	-1	-1	-1
1	-1	-1	-1
-1	1	-1	-1
1	1	1	-1
-1	-1	1	-1
1	-1	1	-1
-1	1	1	-1
1	1	1	-1
-1	-1	-1	1
1	-1	-1	1
-1	1	-1	1
1	1	-1	1
0	0	0	0
$-\alpha$	0	0	0
α	0	0	0
0	$-\alpha$	0	0
0	α	0	0
0	0	$-\alpha$	0
0	0	α	0
0	0	0	$-\alpha$
0	0	0	α

must be determined. Axial points are those values for each design factor which are set when the rest of the design factors are zero (with the exception of the center point). The center points are found by (Myers, R. H., 1976):

$$\alpha = \left(\frac{QF}{4}\right)^{\frac{1}{4}} \tag{3.76}$$

where,

$$Q = [(F+T)^{\frac{1}{2}} - F^{\frac{1}{2}}]^2$$

$$F = \text{number of factorial runs} = 2^4 = 16$$

$$T = 2 \cdot k + 1 = 9$$

Table 3.6 shows the "levels" of the design variables (i.e., 1,0,-1 and α) used for

Table 3.7. Design variable ranges

VARIABLE	UNITS	MINIMUM	MAXIMUM
Primary Mirror Diameter	meters	1.5	2.5
Beam Diameter	centimeters	5.0	15.0
Orbit Radius (From center of earth)	kilometers	8000	42,000
Time Response	seconds	1.0	4.0

data collection and in the regression analysis. The value for α is 1.414. The complete factor design matrix is presented in Appendix G. This scaling of the design factors over their range is discussed next.

3.5.2.3 Selection of Design Variables Ranges The design variables each have a range of possible values. These ranges are listed in Table 3.7.

The primary mirror maximum diameter was selected as that of the largest solid orbiting mirror in existance today; this mirror is on the Hubble Space Telescope. The minimum diameter was selected to be below the limiting resolution of current small mirror satellites (Sky and Telescope Magazine, Apr 90). The largest extreme carries the penalty of high manufacturing costs and high mass but allows for maximum light reception. The lowest extreme produces the mimimal mass for the optical system at a reduced cost, but also restricts the amount of light reception.

The beam diameter contributes to the stiffness of the vibratory modes in the frame structure. The larger the diameter, the stiffer the structure; this quality reduces the magnitude of vibrations in the system. However, an increased beam diameter increases the mass of the satellite as well.

The system is allowed to take on an orbit from low-earth to geosynchronous. A low-earth orbit allows more light to be gathered from the missile, but the satellite may need to slew faster during tracking than if it were farther out. The faster slewing leads to increased system vibrations.

The time response of the control system is slightly more complex to set than simply inserting the value into the control algorithm. This algorithm also incorporated an overshoot value which, along with the time response, needed adjustment to produce a stable system. However, the control model was constructed in a manner such that the settling was a variable which best characterized the control model's performance.

3.5.2.4 Design Variable Scaling Often it is convenvient to scale, or code, the design variable values. This method is particularly useful when the variables cover ranges which differ by several orders of magnitude. Another benefit is that of avoiding the mixture of different units of measure. In the system model, both situations are present. For example, the beam diameter has a range of a few centimeters, whereas the orbit radius range is over 30 million times that of the beam. Also, the first three variables are in distance measures, but the time response is measured in units of time.

When defining a region of interest for a variable x_i, define the range as S_i and the center value as x_{i0}. The scaled value of x_i is represented as:

$$f_i = \frac{x_i - x_{i0}}{S_i} \tag{3.77}$$

This transformation conforms the usual range of design factors to fall between -1 and +1. The CCD design requires model simulation at values of $\pm\alpha$ to determine a second order empirical model. Once optimization of the empirical model is completed, the values for the design variables are obtained by solving the above equation for x_i.

3.5.3 Individual Optimums The completed model was exercised using the factor settings previously specified. For each of the 25 different model simulations, data was collected on the four performance measures. See Appendix G for the raw

performance data for each system simulation. Because the collected data represented different measurements of system performance and no attempt was made to establish specific goals for any of the performance measures, it was decided to optimize the performance measurements in a vector sense. To optimize this system, it was decided to simultaneously minimize mean track error, inverse light intensity upon the primary mirror and system mass while maximizing light intensity upon the detector divided by the image variance.

3.5.3.1 Data Standardization The first step towards analyzing the system performance was standardizing the collected data. It was decided to convert the collected data into unitless values. Each collected performance measure constituted a vector of samples from the performance measure domain. For the matrix of raw performance measures collected, $\omega_{i,j}$, (i=1,...,25, j = 1,...,4), sample means μ_j and sample variances S^2_j were calculated using the following equations:

$$\mu_j = \frac{1}{25}\sum_{i=1}^{25}\omega_{i,j} \tag{3.78}$$

$$S_j^2 = \frac{1}{24}\sum_{i=1}^{25}(\omega_{i,j} - \mu_j)^2 \tag{3.79}$$

Each sample was then standardized into unitless performance measurements, $y_{i,j}$, using the following equation:

$$y_{i,j} = \frac{\omega_{i,j} - \mu_j}{S_j} \tag{3.80}$$

It was then possible to compare system designs and performance without regard to the units of measure. See Appendix G for the transformed performance data.

3.5.3.2 Regression Analysis Least squares regression was applied to find empirical models for each of the unitless performance measures with respect to

the design factors. The software package SAS was employed to perform the regression analysis. The regression analysis resulted in second order polynomial equations estimating the response of the system performance measures. A full second order regression with four factors would normally result in the estimation of 20 individual model parameters. Unitless data has zero mean. Therefore $\beta_0 = 0$ and 19 parameter estimates are required for each model. Statistically insignificant parameters were disregarded resulting in empirical models with four or five factors. Statistical analysis was performed using the resulting analysis of variance (ANOVA) tables to ensure the models provided accurate estimates of system performance and that any parameter estimates were not equal to zero. See Appendix G for regression ANOVA tables. The residual errors for each model were plotted to ensure the models were not biased with respect to predicted values. See Appendix G for residual plots for each model. After examining the ANOVA tables and residual plots, it was decided that the four empirical models provided acceptable estimates of the system performance measures.

3.5.3.3 Individual Optimums Each performance measure is described by a second order empirical model. To find the collective optimum, it was first required to find the optimum of the individual performance measures within the design region. The four unitless empirical models describing the system performance determined by regression analysis are:

$$MTE = -.534017 f_1 + .825842 f_2 - .35130 f_1 f_2 + .45128({f_1}^2 - .8) \qquad (3.81)$$

$$ILI = .835107 f_1 - .536822 f_4 - .407221 f_1 f_4 + .185209({f_1}^2 - .8) \qquad (3.82)$$

Table 3.8. Optimum design factor settings.

Performance Measure	Optimum	f_1	f_2	f_3	f_4
MTE	-1.5295	.41240	-1.414	0	0
ILI	-1.1505	-1.414	0	0	-.1061
IVR	3.3739	-1.414	0	0	1.414
SM	-1.5776	0	0	-1.414	-1.414

$$IVR = -.748866 f_1 + .486955 f_4 - .453307 f_1 f_4 + .600368({f_1}^2 - .8) \qquad (3.83)$$

$$SM = .734666 f_3 + .793570 f_4 + .088546 f_3 f_4 + .129191({f_3}^2 - .80) + .209531({f_4}^2 - .8) \qquad (3.84)$$

where,

f_1 = System orbit radius (standardized)

f_2 = Control system settling time (standardized)

f_3 = Structure beam diameter (standardized)

f_4 = Primary mirror diameter (standardized)

$-1.414 \leq f_i \leq 1.414$

These equations can be displayed in graphical plots, since each are functions of only two factors. See Appendix H.

The Automated Design Synthesis (ADS) FORTRAN program was used to find the individual optimum to the above empirical models. The individual optimum and corresponding factor settings are listed in Table 3.8.

The design settings for the optimization of SM and ILI were predictable. The behavior of MTE and IVR was not as easily forcasted because of the complex nature of the response. It was not known to what degree each of the design factors

would contribute to these performance measures. After finding the optimum for each performance measure, the following comments can be made about the optimums. The minimum MTE was found at the lowest control system settling time which was expected but the best orbit was found to be closer to geosynchronous than low Earth orbit. Apparently, the higher the orbit, the easier it is for the system to track the missile. With regard to ILI, it is well known that the intensity of a light source falls off inversely with the square of the distance to the light source. Considering that, it is expected that the minimum orbit radius will provide the highest light intensity. One would also expect to maximize light intensity upon the primary mirror by maximizing the area of the primary mirror. This is not the case, however. ILI is optimized when the primary mirror is rather small. This is due to the relatively small change in the range for the primary mirror size and added vibrations caused by increasing the mirror size. IVR is optimized at expected factor settings. The orbit radius is at a minimum while the primary mirror diameter is at a maximum. The optimum primary mirror size provides an insight into the image on the detector plate with respect to the primary mirror dimensions. It appears a large primary mirror size provides higher image quality. SM optimum occurs at expected factor settings, minimum beam diameter and minimum primary mirror radius.

Two of the design variables, time response and beam diameter, were each only significant in one of the performance measures. The time response was significant only in the MTE and the beam diameter only in the SM. This result eliminated these two variables as contributors in the global system optimization to follow; they would take on minimum values in the minimization of their respective performance measures. This process would be due to the fact that they had no coupling with any other performance measure. Nevertheless, the remaining two variables coupled with the performance measures. Also, they took on different values for the separate performance measure optimums. This result required a simultaneous optimization of all the performance measures.

3.5.4 Simultaneous Optimization Provided with empirical models and their optimums for each of the four performance measures, a method was required to select the best design of the system while considering all performance measures. The method by which to optimize multiple responses of multiple, controllable factors (Khuri, A. I. and Conlon, M., 1981) involved 'simultaneously' optimizing through a minimization of a distance function ρ.

Given responses to the system model linear model for a given j^th response can be expressed as:

$$\vec{y}_j = X\theta_j + \vec{\epsilon}_j \tag{3.85}$$

where $\vec{y}_i$ is a vector of p observations, $\vec{\theta}_i$ is a vector of m unknown design factors, and X is an p by m matrix of constant coefficients. If there were r responses (i.e. measured system perfomaces), then the model would be:

$$Y = X\Theta + \epsilon \tag{3.86}$$

where Y and Θ are now in matrix form with their columns formed by partitions of the r vectors corresponding to the i^{th} design. ϵ is a p by r matrix of errors

An important note: When referring to y_{ij}, i refers to the design of the model and j refers to the specific response number (out of a total of r) for that simulation. An unbiased estimator for the variance-covariance matrix of Y is the matrix $\hat{\Sigma}$:

$$\hat{\Sigma} = Y^T[I_n - X(X^TX)^{-1}X^T]Y/(n-p) \tag{3.87}$$

An important consideration of this formulation is that if $r \leq p - m$, then $\hat{\Sigma}$ will be nonsingular.

A check on possible linear relationships between the responses is performed; obviously if a linear relationship exists between any two response, then their global optimum is completely determined by either one of the separate response responses optimums. If these dependencies are present, then one or more of the dependent responses can be removed. The mechanism by which these dependencies are revealed was the DD^T matrix, where the ij^{th} component is the j^{th} response corrected (subtracted) by its mean. In matrix form,

$$D = Y^T(I_n - 1_n 1_n^T/n) \tag{3.88}$$

where 1_n is an n vector of ones. If an eigenvalue of DD^T is zero, then there is a linear dependency among response types. There is also a check on the round-off error of the numerical data to see if this type of error affects the DD^T eigenvalues. The round off error is assumed composed of independent identically distributed random variables and its standard deviation is checked against the smallest eigenvalue of DD^T. If an eigenvalue is within, say, three sigma of zero, then it is assumed to be a possible zero eigenvalue.

The estimate of the responses in matrix form is given by:

$$\hat{Y}(\vec{x}) = \vec{Z}\,(\vec{x})\hat{\Theta} \tag{3.89}$$

where $\vec{Z}$ is a vector of p design factors in the form of $\vec{x} = \{x_1, x_2, \cdots, x_{p-1}\}^T$.

In this research, a second-order model is desired; therefore, some of the coefficients are quadratic combinations of the design factors. An unbiased estimator of the variance of $\hat{Y}(\vec{x})$ is:

$$\mathrm{Var}[\hat{\mathrm{Y}}(\vec{\mathrm{x}})] = \vec{\mathrm{Z}}_{\mathrm{T}}\,(\vec{\mathrm{x}})[\mathrm{X}^{\mathrm{T}}\mathrm{X}]^{-1}\,\vec{\mathrm{Z}}\,(\vec{\mathrm{x}})\hat{\Sigma} \tag{3.90}$$

The individual response models, $\hat{y}_i(\vec{x})$ are each optimized separately to produce individual optimums, ϕ_i. If each of the optimums occur at the same values for $\vec{x}$, then the optimization is finished. This event is extremely unlikely, especially if the individual optimums tend to drive the design factors towards opposite extremes. The value of the simultaneous multiple response approach now becomes apparent.

The following premise is established: A single optimum is desired to be as close as possible to all the individual optimums. Therefore, form an r-dimensional Euclidean space with the ϕ_i and define r as the distance from $\hat{Y}(\vec{x})$ to the vector of individual optimums, Φ.

There are several choices for this distance function, but the one chosen for use in this research is:

$$\rho[\hat{Y}(\vec{x}), \Phi] = [\sum_{i=1}^{r}(\hat{Y}(\vec{x}) - \phi_i)^2/\phi_i^2)]^{\frac{1}{2}} \tag{3.91}$$

The function is simply a type of regression on $\hat{Y}(\vec{x})$ where the mean is represented by Φ. It is an adjusted squared mean which is divided by the optimum value for that specific response type.

This measure is a very valuable tool when the responses, $y_i(\vec{x})$ are each measured in different units or are scaled to different orders of magnitudes.

Under ideal circumstances, this distance measure should account for the randomness of Φ by establishing a confidence region about the values of $\vec{x}$. Each ϕ_i is actually a random value in itself, due to the estimation of Y. However, this confidence region restricts the range of $\vec{x}$ over which the analysis can be performed. This research required an analysis over the entire range for $\vec{x}$. To restrict the search over small regions would require more time than the research group has available. Consequently, the distance function, $\rho[\hat{Y}(\vec{x})]$, was optimized after insuring a good fit for $\hat{Y}(\vec{x}, \Phi)$. No confidence interval for ρ was established, ρ being assumed valid over the entire range of $\vec{x}$.

Table 3.9. Optimization results

min ρ	.83411
f_1	-1.2708
f_2	-1.414
f_3	-1.414
f_4	.099513
Orbit radius	9,723,840 meters
Settling time	1 second
Beam diameter	5 centimeters
Primary mirror diameter	1.5352 meters
MTE	-.39158
ILI	-.8003774
IVR	2.0269969
SM	-.7119192

3.5.5 Best Design Once the individual optimums were found, the ρ vector was formed and ADS was applied to find the smallest value over the range of values of f_1, f_2, f_3, and f_4. The results of the optimization are listed in Table 3.9.

The values for the individual responses are derived from the empirical models. The following observations can be made about the optimal system configuration. Optimum orbit radius was affected by three of the four performance measures, resulting in a value in the middle of the design space. The same thing occurred in determining the optimum primary mirror diameter. Settling time and beam diameter optimums were affected only by one performance measure pushing each factor to the edge of the design space. In general, the combined optimum system provides performance close to the individual optimum systems. The relatively low Earth orbit reflects the need for the shortest practical distance between the missile and system. This is caused by the need to maximize the amount of light incident to the detector plate. If this system were just a single satellite in a space defense constellation, a low Earth orbit for the constellation would mean more satellites would be required to cover the area of interest. A complete defense system configuration has not been considered here.

Therefore, the orbit radius optimization was not affected by the number of satellites required to perform the mission. If the complete system was considered, it would undoubtedly drive the system to a higher orbit.

The time response was minimal, since it was only a significant factor in one of the minimized performance measures. Nevertheless, this system should have as fast a control system as possible within constraints. The control system will be limited by the size of practical controllers. Of course, the system should be stable with the control system attached. Also, this design research has not examined the amount of stress placed upon the structure of the system due to the control action. It would be possible to perform a strain-stress analysis upon the chosen design to find the limits of applicable control. Once the control constraints have been determined, a control system should be chosen to maneuver the system as quickly as possible to acquire the missile.

The beam diameter was minimal for the same reason as the minimal time response: it was significant only in the minization of the system mass performance measure. It was expected that a small beam diameter would have a detrimental effect upon the image quality by causing vibrations in the optical system, but the regression analysis did not support this hypothesis.

The optimum primary mirror diameter has been influenced by the need to acquire a sharp image of the missile while also needing to reduce weight. As the primary mirror surface area increases with its radius, so is the mass increased. The midrange setting of this design variable illustrated the conflict of keeping the system mass at a minimum, yet maximizing the light collected by the primary mirror surface.

IV. COMPARISON OF METHODS

4.1 SUBSYSTEM LEVEL DESIGN AND ANALYSIS

The development of a complex system traditionally requires a division of labor within a design group. Historically, the group members are separated into smaller working teams in accordance to their technical specialties. These teams are then required to develop a subsystem which most closely fits their expertise. The development of these subsystems requires simulation models to some level of detail. Engineering modeling, up to the middle of this century, generally required a set of well-used formulas and a large set of initial assumptions to produce a first prototype. The prototype would be tested and the design flaws would be corrected in the model. Further analysis, prototypes, and testing would follow. This design process could involve many iterations; furthermore, with separate design teams interfacing as previously mentioned, the cost and time required for development could grow quickly. With the recent advent of computers, however, the testing and model modifications can now be accomplished in complex computer models. Sweeping assumptions to simplify the model need not always be implemented, thus allowing quick and accurate modeling.

Computer modeling, while alleviating many system engineering burdens, emphasizes others. The specialization of engineering disciplines created whole new languages and perceptions of how simulations should operate. For example, the primary concern of an electrical engineering team could be the mathematical value of the plant to be controlled; yet, they may have only the vaguest notion of how to obtain this value or of its sensitivity to other subsystems. A mechanical engineering team may be providing this plant value to the previous team; this team could be similarly unaware of the plant's affect on the control subsystem. As the complexity of systems increases, the number of subsystems and design teams increase; this es-

calation in the number of engineering projects requires structured methods for the models—as well as the engineers—to interface effectively with each other.

Several project management techniques are available to manage the problems listed above. At least one member from each engineering team is selected to form an integration team. Information on the input and output of their different subsystem models are coordinated at the integration team level. Another variation of this integration team is to compose its members from outside the separate engineering groups. This team would be uninvolved with the details of the separate subsystems; they would be primarily concerned with the passage of information between the subsystem models. The last approach is to have no division of the design group; instead, the group is given the task of designing the entire system. They must develop it as one unit without concern about integrating their results with anyone else.

These management methods have parallels in the approaches used to handle model integration. Three of the most promising are the unified computer language approach, Integrated Structural Modeling (ISM), and bond graphs.

The Jet Propulsion Laboratory is investigating the coupled effects of structural vibrations against highly precise optical ray tracing (Briggs and others, No Date). Their models are all exclusively developed in one, unified, computer modeling language: MATLAB. MATLAB is a matrix-manipulating computer package which has many engineering-oriented subroutines. The philosophy of using one language to handle the entire simulation is analogous to the management approach of not separating a group into design teams. The individuals of the group will be knowledgeable about the entire system simulation; to do this, the expertise of these individuals must be extremely high in nearly all the necessary engineering disciplines.

This technique of system integration has several benefits. It requires all the functions of the system to be modeled in similar detail. There is no concern about exporting or importing data for the simulation, since all of its requirements are

handled internally. The model can be used throughout an entire design process by continually adding detail to the system model. Even though the unifying computer language technique has merit, there are several drawbacks. The primary difficulty, from a system engineering viewpoint, is that the system simulation model is developed without great regard towards external integration with other system models. This trait is often undesireable in a model; another design team may wish to easily build upon the original model by transporting its functional I/O to other popular engineering simulation packages. Also, there is the danger of completely submerging the subsystem models into the global system model. This merging process makes simple structural changes in a subsystem model difficult to accomplish without affecting the entire system.

Lastly, some models may require intensive effort to simulate in a single code. MATLAB is a very versatile programming tool, but there are many problem-specific tools available that perform certain simulation more efficiently. For example, NASTRAN is specifically developed for finite element structural analysis. A tool or technique which incorporates different, separate models could be more beneficial for system modeling.

A computer package which is specifically designed to interact with external simulation packages is ISM. The function of ISM is transform the output and input files of these separate packages into its own format. ISM is then used to manipulate this data and/or translate the data between the separate models. These models can reside in MSC/NASTRAN, COSMIC/NASTRAN, ANSYS, MatrixX, EASY5, BDISCOS, TOPS and others. Some of these models are external to the ISM integration program; others are internal (see Figure 4.1). This philosophy of computer management of models is similar to the engineering management technique of using a separate system group coordinate the specific engineering design teams.

The satellite system could be modeled using ISM. The structure model would be exercised in NASTRAN, the controls model in MatrixX, the orbital and ballistic

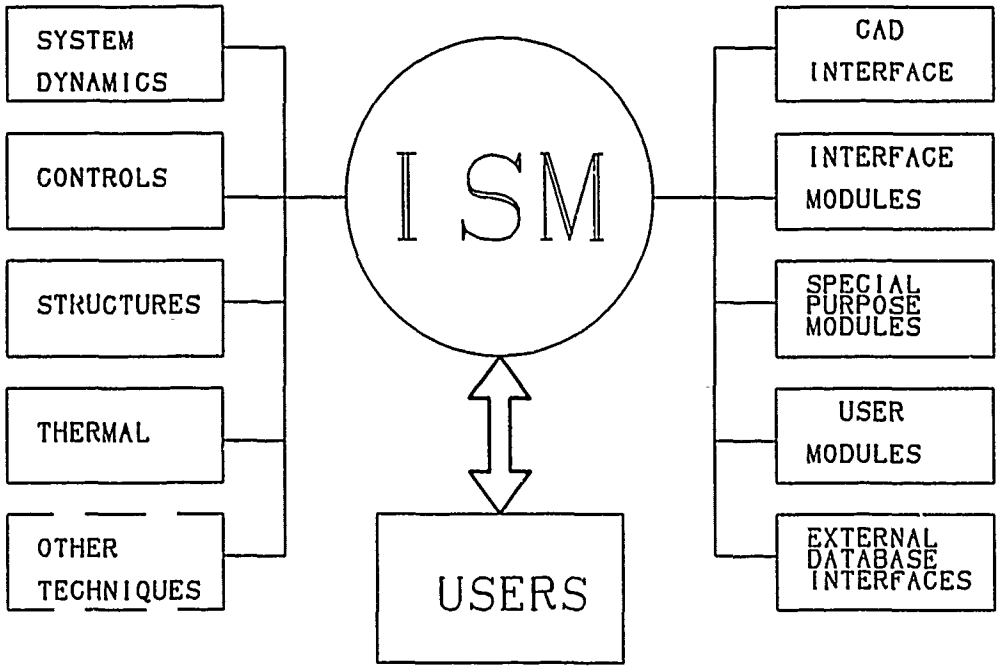


Figure 4.1. ISM schematic.

model in BDISCOS, and the optical model in COMP or TOPS. The bulk of the structural model in Chapter 3 was prepared in NASTRAN, so implementing ISM for the structure is straightforward. Additionally, the optical system model was based on the same fundamental methods as in COMP; therefore, it could be implemented with some additional effort. The controls, orbital and ballistic missile models would have to be completely programmed from the beginning; controls in MatrixX, orbital and ballistic in BDISCOS.

A benefit of ISM is its simple concept. A system works internally through its input/output (I/O) exchange between subsystems; ISM functions as a facilitator for the process. In theory, ISM allows an engineer to retain his models in a simulation package best suited for his needs and still allows for integration with other dissimilar models and codes. The separate models are independent; changes in their construction do not affect the rest of the total system model as long as the I/O format is correct. If indepth, numerical manipulation is necessary between models, ISM has in-house programs which can accomplish many transformation tasks.

There are many disadvantages to using ISM and similar techniques. Computer management solutions tend to require a high proficiency in computer network and programming skills; ISM is a prime example of this requirement. Its commands are complex, difficult to understand and implement. ISM communicates effectively with computer programs, but its communication with humans is very poor.

The passing of data between models in computer-oriented integration programs has a fundamental flaw: the data are merely numbers to be used; they have no meaning to the computer package. This flaw is nonexistent in bond graph theory and therefore is a much smaller problem in the computer implementation of bond graphs.

Frequently a design group wants to optimize a system through active change of the subsystem variables. Using separate codes for different models usually requires each model's parameters to be changed by accessing them through their specific parent simulation codes. This process is very inefficient. In general, using several different simulation codes to simulate one large system is not efficient. An alternative is to have separate models, but retain them in a single simulation language or package.

The bond graph method is a single theory which can incorporate widely different models. Bond graphs treat all engineering models as subsystems of power flows; thus, dissimilar engineering disciplines can be directly linked in an overall system model. The computer package ENPORT uses the bond graph theory to link different models together to form a single system, but additionally allows input from other models which are not modeled as bond graphs.

The use of bond graph modeling in system development is analogous to the use of the first integration team mentioned above in system management. Here, an integration tool is used that can go into detail in a specific discipline, interface only peripherally with another and still retain the aspects of the total system. Bond graph construction produces a graphical road map (with or without a computer simulation

package) from which a system engineer can clearly define the interface connections. And most importantly, the theory requires all input/output to be expressed explicitly in terms of power. This requirement stems from the fact that this integration method is a theory and not just a design approach. Thus, bond graphs address the problem of interpretin' ubsystem model interactions by building the power flow requirement into its theory.

Bond graphs have been used in a wealth of different fields. At first, the technique was primarily used to model distributed, lumped systems, but have since been used to describe engineering problems ranging from structural modal analysis to pipeline transients and even to economic systems(Bos and Breedveld, 1985). All of these fields of application center on the premise of power flow, and consequently can all be described in the same language. This simple fact is a great aid in communication between different engineering communities.

The flexibility of bond graphs as a modeling tool is illustrated by investigating areas of interest for the satellite system which were not fully modeled in the research. The four areas are: flexible optics, thermal effects, passive vibration control, and active vibration control.

4.1.1 Flexible Optics The design and development of flexible optical space platforms will undoubtedly require a vibration sensitivity analysis. The system model contains the ability to determine the effects of vibrations on the optics as a system. However, the model does not allow displacements between the different optical components. A more accurate model of the system may be achieved by allowing motion between the different optical components.

The system model adapts easily to this change. The vibrations model contains all the dynamical equations of the structure. If the optical structure is allowed to become flexible (instead of rigid), the structural vibrations model must be changed. The original structures model was based on the NASTRAN analysis of the system.

A new system would be modeled, adding a flexible subsystem of bea.. to connect the optical components.

4.1.2 Thermal Effects It is possible for th.. effects of thermal radiation to be a factor in the design of space based optical systems. The effect of structural heating due to thermal radiation manifest itself as a displacement in unconstrained beams. Temperature displacements in the beam may be due to either uniform changes in temperature or to differential changes. A uniform change refers to a temperature change that is constant throughout the beam. A uniform change causes the beam to increase or decrease in length. All other dimensions of the beam will change proportionately, but only the change in length is significant in most cases. A differential change means that the top and bottom of the beam are subjected to different temperatures, while the average temperature remains unchanged. The member will not change in length but will undergo a curvature of its longitudinal axis.

With an understanding of the thermal environment of the system, both the uniform temperature change and the differential change in temperature can be defined for each beam as functions of time and/or other system parameters. The primary effects of structural heating due to thermal radiation can then be incorporated into the model using the existing modulated transformers.

4.1.3 Passive Vibration Control In the current system model, no attempt was made to isolate the optical assembly from vibration in the outer structure. The system, including the optics, was modeled as one body. The orbital dynamics model and the rigid body rotational model combine to model the rigid body characteristics while the structural vibrations model captures its flexible characteristics. However, the system model can be expanded to include some form of passive vibration control at the connection points. The most effective way to add passive vibration control is to model the outer support structure and the optical assembly as two bodies connected by spring and damper assemblies. In doing so, the bond graphs that modeled the

dynamics of the one body system would be duplicated and added to the original model to form a two body system. An additional structural vibrations model would be added to captures the flexible characteristics of the optical assembly.

4.1.4 Active Vibration Control One large item of interest in the control of large flexible space structures is active control of structural vibrations. Once again the problem of not having a fixed frame of reference from which to apply forces is present. Thus any attempt in actively controlling vibrations will result in the actuator being mounted on the structure itself. The difficulty is that when the structure moves, the actuator moves with it, resulting in dynamic coupling between the actuator and the structure. In order to design an effective control algorithm, this coupling must be accounted for. Many research projects are involved in modeling this coupling and determining optimal placements of the actuators on the structure as well as sensor placement.

The techniques utilized in this study can be easily extended to active vibration control using linear proof mass actuators.

The bond graph of the proof mass controller is shown in Figure 4.2. The applied force to the proof mass is supplied by an electromagnetic motor. The electrical current in the primary windings is converted to a mechanical force on the proof mass through the magnets in the ring. The conversion from a flow signal to an effort signal is accomplished with a gyrator. The modulus for the gyrator is equal to the force constant of the motor.

Integrating this bond graph with the system is shown in Figure 4.3. The active controls model interfaces with the vibrations model. The modal acceleration of the structure can be easily determined from the forces applied to the modal masses. The modal accelerations are then transformed to generalized accelerations which are then input to the proof mass controller. In return, the resulting active control forces upon the structure must be sent through transformers to get modal forces. The resultant

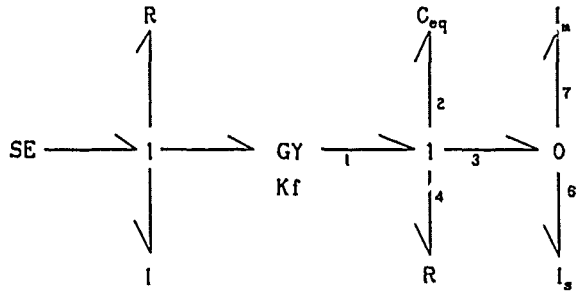


Figure 4.2. Proof mass controller bond graph.

modal forces, due to active vibration control, are summed with the modal forces due to momentum wheel control. The summed modal forces are then applied to the vibrations model to determine the motion of the flexible structure.

4.2 SYSTEM LEVEL DESIGN AND ANALYSIS

The system design and analysis process was that process involved in the development of the entire satellite system. In the simulation arena, this process can be divided into three general activities: Individual models, system integration and system optimization. The comparison of conventional modeling and analysis methods and those of bond graph methods will be divided among these three different activities.

4.2.1 Individual Models

4.2.1.1 Conventional Methods The use of specialized computer simulation packages in the conventional methods have been mentioned above. These codes can become quite complex, depending on the discipline to be simulated. For example, the satellite structural dynamics model was modeled in MSC/NASTRAN. The structure was simple in concept and had few finite elements defined. However, NASTRAN was quite difficult to learn; in fact, another computer package, SDRC/IDEAS, was used to develop the dynamics structure. The NASTRAN code was then written by IDEAS. In other words, a computer translator was necessary to

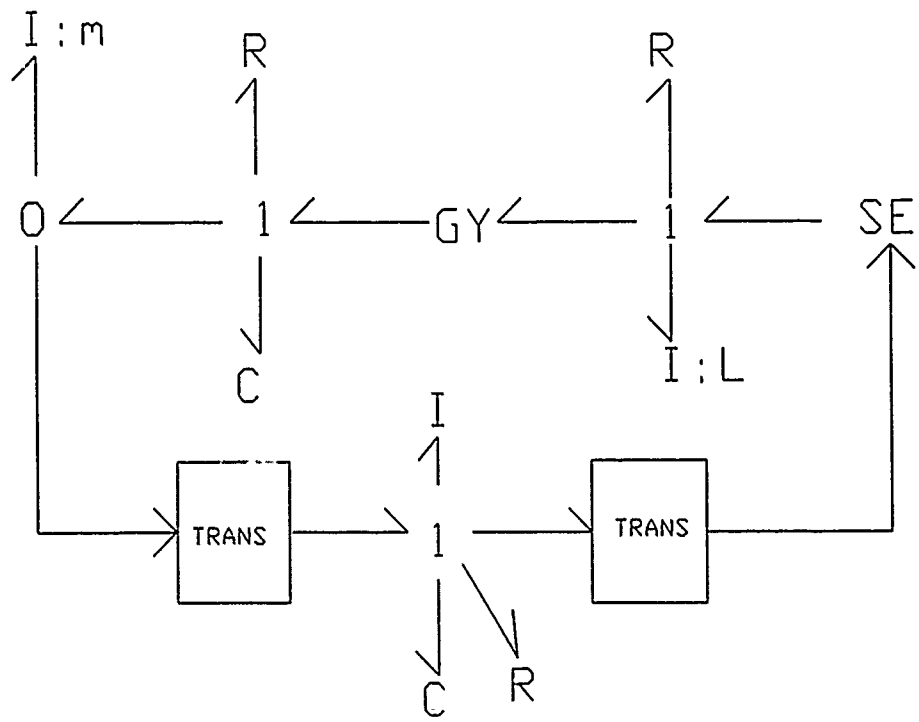


Figure 4.3. Controller/structure interface.

communicate with the modeling computer package. Though other modeling packages may be more user-friendly, they also have similar complexities. The advantage of these packages, however, is that they can handle very detailed simulations.

Subsystem models of different disciplines traditionally require different modeling techniques and simulation packages. A structural dynamics model would not normally be modeled in the same package as that of an orbital dynamics model. Consequently, a separate technique or computer package must be learned for each subsystem. The six different models of the satellite system, for example, must be modeled in four or five separate computer simulation packages.

One advantage of the traditional use of computer simulation packages has been in the advance of graphical output capability over the last two decades. Some simulation packages can produce highly detailed and very helpful graphical outputs to the user. These outputs are usually in the form of screen or hardcopy graphics. Some of

the common ones include PATRAN, MATRIXx, IDEAS, and others. However, this graphical ability is not available for all disciplines nor is it standardized. The quality and indeed the meaning behind the graphical output varies from one specialized package to another.

4.2.1.2 Bond Graph Methods The bond graph method produces subsystem models which are entirely complete in their descriptions. The purest output of the bond graph analysis is a set of state equations which describe the model's behavior; state equations are very manageable from a numerical point of view.

The bond graph method crosses many different engineering disciplines while using the same exact terminology. A bond graph used for electrical circuits has the same symbology and notation as a bond graph model for a structural model. One theory and one language may be learned for a variety of different models.

Bond graph theory provides a useful and intuitive graphical interface. The graphical output is more than the visual display of the simulation result, as in conventional simulation outputs. There are graphical maps which vividly display the interconnections inside and between subsystems. These interconnections are not just hypothetical connections on a flow chart; they represent physical power flow which has been quantified. In other words, bond graph diagrams are not merely output displays; they visually represent the actual construction of the system and its subsystems.

4.2.2 System Integration

4.2.2.1 Conventional Methods After developing the separate subsystem models, they must be integrated to form a single, complete, system. Conventional modeling methods have some real difficulties here.

The separate, subsystem models are often constructed independently and in different computer simulation packages. Consequently, the output and input, or

I/O, requirements of the different models may not be compatible. Some models, while good for a specific analysis in a particular area, simply may not be suitable for integration into a system. For example, a structural engineer may develop a model for dynamic analysis which is only half of a symmetric structure. This is entirely reasonable, but integrating only half of the support structure into a total system simulation would not be acceptable.

There are also computer integration problems. Often, the connection of different subsystem models, housed within separate computer simulation packages, require manual data reformatting and transfer between these models. When computer punch cards were used years ago, this process meant a person literally typed new input cards and carried this new data to the next simulation computer. Today, this information flow between subsystem models is accomplished electronically, but human interface is still required. In addition, a high level of computer experience may be required to connect the different simulations together. Though these problems seem to be a computer programmer's problem, the system engineer is the person who usually must solve them in order to have an effective system simulation.

4.2.2.2 Bond Graph Methods System integration of models is where the bond graph method really shows its power.

The use of bond graphs force the modeler to anticipate the future integration of his model into a larger system. This is because all input and output information must be in the form of power flow, or perhaps in signal flow. Thus, the system model is formed by using bond graph connections between the subsystems. There is no incompatibility problem concerning the transfer of data between subsystems, either from a conceptual or programming standpoint.

One more important aspect of bond graphs was made apparent during the satellite model development. A discipline which may not be readily adaptable to the bond graph theory can still be integrated into a bond graph system model. This is

possible because bond graph modeling allows for block diagrams in signal processing. In this case, modeling an exact ray-trace model of the optical system was possible only by creating a FORTRAN program. However, the optical model was restricted to importing and exporting in terms of signal flows; specifically, in terms of positions. This allowed direct integration of the FORTRAN simulation into the entire bond graph system model.

Bond graphs facilitate a complete model simulation. The subsystem models are connected by power bonds to form a larger system model. The amount of power in a bond is determined by the state of the energy variables at any given instant in time. In the satellite system model, the control subsystem model is connected to the satellite rigid body model and the structural vibrations model via a single 0-junction. The control power is distributed into the bonds of the separate models based upon the state of the satellite system model's energy variables. In general, there is a natural coupling between subsystems information flow. The subsystem interactions are readily identifiable, since all the flows are displayed in a graphical nature right down to the subsystem level.

4.2.3 Optimization

4.2.3.1 Conventional Methods The process of system optimization requires system-level performance measures to be collected. This collection process consists of varying the subsystem variables values at specific levels and then collecting the data for analysis. The conventional system model is a connection of disjoint, specialized computer simulations. Performance measures are often difficult to collect in such a system. In addition, a design variable change in one subsystem model may have its affect isolated in that one simulation. The system-wide effect of this variable change, however, may need to be more wide-spread.

Search procedures, such as the Golden Search technique, are commonly used to optimize simple models towards an optimum performance level. A system composed

of disjoint, specialized subsystem models, however, may require intensive manual data transfer during a single optimization iteration. With a search method requiring hundreds, even thousands, of optimization iterations, this technique is clearly impossible.

4.2.3.2 Bond Graph Methods The bond graph of a system is a complete, unified model. It can be manipulated and examined as a unit, or investigated at lower levels if desired. System performance measures are therefore collected from any system level or from any subsystem. When variable changes are made during the data collection process, the effect of this variable change is immediately distributed throughout the system via the power bonds.

Search procedures are very simple to implement into a bond graph model. Since the system is entirely described in a single model description, attaching a search algorithm will make the optimization of the system an automated process.

The system analysis process often may involve major changes to the system model or the subsystem models; it may also require new subsystems to be added to the current model.

4.2.4 Flexibility

4.2.4.1 Conventional Methods After a system model is analyzed at a specific level, various subsystem models will probably be increased in detail. Conventional subsystem models, in their specific computer modeling packages, are built to allow increased detail. A new subsystem model may be required, but the foundation for its construction will have been laid by the simpler model.

However, the addition of a new subsystem model to a current system model results in the same integration problems as earlier detailed. The new subsystem model may have compatibility problems with the original model. If the new model

is implemented on a computer simulation package different from the ones previously used, then there is the additional problem of learning another language or discipline.

4.2.4.2 Bond Graph Methods Bond graphs can go into greater detail for subsystem or system models. The structural vibrations bond graph allows for a simple increase in the number of modes used in a given analysis. This increase is accomplished by simply adding another 1-junction at the top of the modal bond graph "tree"; the appropriate power bonds and transformers are added from each 0-junction. The structural vibrations model also shows how bond graphs may lose their graphical simplicity as detail is increase. This problem results in all the power flow interconnections becoming woven into a confusing maze. The problem appeared on an early effort in this research. Bond graphs were used to simulate structural finite elements; with less than twenty elements in the structure, the structural subsystem model became unreadable. This initial setback inspired the use of modal analysis for the structural dynamics problem. The modal analysis method allowed for a more simple bond graph, yet still allowed for an equal degree of accuracy.

The bond graph system model can easily handle a new subsystem model. If the new model is in a bond graph form, then the power and signal flows merely need to be added to the appropriate junction structure. If the new subsystem model is not adaptable into a bond graph form, then it must merely conform to power or signal I/O. The integration of this new model would be accomplished in a manner similar to that of the optics subsystem model used for the satellite model.

There were specific expansions/adaptations which were investigated for the satellite model, but were not added as part of this research (see Appendix I). These expansions/adaptations handled thermal effects, flexible optics substructure, and passive and active vibration control. These expansions/adaptations exhibited the flexibility of the current satellite system model to incorporate additional subsystems.

V. CONCLUSIONS

Bond graphs and Systems Engineering methodologies were found to be an extremely powerful conceptual design tool. Bond graphs allowed the integration of several different models while Systems Engineering methodologies were very good at defining the interfaces between the different models.

The initial construction of the individual models was easy and resulted in simple models that accurately described the dynamics present in the system. Traditional techniques invariably result in overly complex subsystem models. The complexity of these models hinders the design engineer in the determination of the important system variables. These variables are the ones that have the greatest effect on the system performance. By defining the important system variables, the design engineer can focus his attention on these variables and base early design decisions on these variables.

Bond graphs allowed for integration of several subsystem models involving different engineering disciplines into one complete system model. This was accomplished routinely. Traditional methods provide an extremely difficult, if not impossible, method for integrating several models together. Without a complete system model it is virtually impossible to determine how the dynamics of the complete system will react as a function of time. Because bond graphs allowed for this integration, the complete system dynamics could be seen. Any changes made to the system model was immediately seen in the total system performance, thus allowing the design engineer to see immediate results.

Bond graphs were also shown to be flexible in that they can be quickly modified to reflect design changes. The addition of more subsystem models to the system model is quickly accomplished by modeling the power flow between the systems. Thus the entire system design can be changed by adding a new susbsystem and immediately affect the overall performance of the system.

A complete system model was created, thus allowing for optimization of the system as a whole. Traditional techniques focus on the optimization of individual models because optimization of the system as a whole is nearly impossible. However, because the entire system was represented by a single model in bond graphs, optimization of the complete system was shown to be possible. This optimization focussed on the important system variables and the effects they have on the system performance.

As an advanced design tool, bond graphs were also seen to have an important role. Because changes to the bond graph are done routinely, the entire system model can be modified as the design evolves in the advanced design stages. Thus a complete system representation is maintained throughout the design phases allowing for quick determination of the effects a design change will have on system performance.

Bond graphs are clearly the system modeling choice during early conceptual design phases. This is due to the fact that simple models that accurately define the dynamics of the system can be built quickly. During advanced design stages, the bond graph models built in the early design phases can be maintained and updated easily allowing for the complete system representation to be maintained.

Appendix A. *MISSILE DYNAMICS MODEL*

A model of the missile dynamics was required to determine system design parameters. The model was also used to provide inputs to the system during performance evaluation. Three models were developed. A math model was first developed and then used to develop both a SLAM and bond graph model.

A.1 MATH MODEL

The missile math model was developed with the assumption that only two forces act on the missile, gravity and thrust. The missile trajectory was also assumed to be planar with respect to the earth frame.

A.1.1 Gravity The gravitational force on the missile is:

$$F = \frac{GMm}{R^2} = \frac{\mu_e m}{R^2} \tag{A.1}$$

where,

F = Gravitational force

M = Mass of earth, kg

G = Gravitational constant, $\frac{km^3}{kg-sec^2}$

μ_e = GM = 398,601.2 $\frac{km^3}{s^2}$

m = mass of missile or satellite, kg

R = distance of object from center of earth, km

We associate $F = ma_{gravity}$, so $a_g = \frac{\mu_e}{R^2}$. The direction of this acceleration is always towards the center of the earth. One way to represent this is by using a cartesian coordinate system on the earth frame. Each component of the acceleration, a_x and a_y, is along their respective earth frame unit vectors. A programming method for determining a_{gx} and a_{gy}, is by taking the components of the position vector $\vec{R}_{x,y}$,

dividing them by $|\vec{R}|$, and multiplying the product by $|\vec{a}_g|$:

$$a_{gx} = a_g \cdot \frac{R_x}{|R|} = \frac{\mu_e}{|R|^2} \cdot \frac{R_x}{|R|} = \frac{\mu_e R_x}{|R|^3} \tag{A.2}$$

where $|R| = \sqrt{R_x^2 + R_y^2}$. Similarly,

$$a_{gy} = \frac{\mu_e R_y}{|R|^3} \tag{A.3}$$

A.1.2 Thrust If an object has a constant thrust, an acceleration due to that thrust a_T , can easily be determined. If the mass of the missile is constant, then:

$$a_T = \frac{THRUST}{TOTALM} \tag{A.4}$$

where TOTALM is the total mass. But, the thrust is produced by a loss of mass, therefore:

$$a_T = \frac{THRUST}{TOTALM - MFR \cdot TIME_S} \tag{A.5}$$

where MFR is the mass flow rate of the stage and $TIME_S$ is the elapsed time during stage burn. The mass flow rate is a function of the thrust and the exit velocity of the propellant (V_e) then:

$$MFR = \frac{THRUST}{V_e} = \frac{THRUST}{I_{sp} \cdot g} \tag{A.6}$$

where I_{sp} is the specific impulse of the engine and g is the acceleration due to gravity at sea level. The maximum duration of the stage burn (TSB) is a function of the propellant mass (PROPM) and the mass flow rate where:

$$TSB = \frac{PROPM}{MFR} \tag{A.7}$$

The duration of the acceleration due to thrust for each stage (maximum value

of TIME$_s$) has a range from 0 to TSB. This duration can be altered for each stage to control the missile trajectory. The acceleration of the missile due to thrust is assumed to be in the direction of the velocity vector. In a manner similar to that for a_g, the components of acceleration due to thrust are:

$$a_{Tx} = \frac{THRUST}{TOTALM - MFR \cdot TIME_S} \cdot \frac{V_x}{|V|} \tag{A.8}$$

$$a_{Ty} = \frac{THRUST}{TOTALM - MFR \cdot TIME_S} \cdot \frac{V_y}{|V|} \tag{A.9}$$

where,

$$|V| = \sqrt{V_x^2 + V_y^2} \tag{A.10}$$

The final math model of the missile dynamics is then:

$$\dot{x} = V_x \tag{A.11}$$

$$\dot{y} = V_y \tag{A.12}$$

$$\dot{V}_x = \frac{-\mu_e \cdot R_x}{|R|^3} + \frac{THRUST}{TOTALM - MFR \cdot TIME_S} \cdot \frac{V_x}{|V|} \tag{A.13}$$

$$\dot{V}_y = \frac{-\mu_e \cdot R_y}{|R|^3} + \frac{THRUST}{TOTALM - RMFR \cdot TIME_S} \cdot \frac{V_y}{|V|} \tag{A.14}$$

Neither the thrust nor the total mass is continuous. The thrust changes discretely at ignition and burnout. The missile mass changes discretely at each stage separation.

A.2 SLAM MODEL

Because the missile math model contains both discrete and continuous events, it was first modeled using SLAM. The SLAM network diagram is shown in Figure A.1. The continuous aspects of the dynamics were modeled using Equations A.11

through A.14. The missile was modeled as a two stage rocket in two dimensions. The flight path can be altered by adjusting the following three parameters:

LCHA = Initial launch angle

WEIGHT1 = Propellant burnt by first stage

PWEIGHT2 = Propellant burnt by second stage

By adjusting PWEIGHT1 and PWEIGHT2, the burn time of each stage is altered but not the total weight of the missile. The total weight is constant. The model was used to simulate the launch of a missile from Kansk, USSR to Dayton, OH. The simulation was made using data for a Titan III missile (NSSA, 1973).

Data was collected on the position, velocity, and acceleration of the missile. The SLAM model was verified by checking the missile trajectory, missile velocity at second stage burnout, and missile position on impact. The SLAM model variables that are readily available include:

Input Variables

1. Weight of first stage
2. First stage propellent burnt
3. Weight of second stage
4. Second stage propellant burnt
5. Weight of reentry vehicle
6. Launch angle
7. First stage thrust
8. Second stage thrust

Output Variables

1. X position
2. Y position

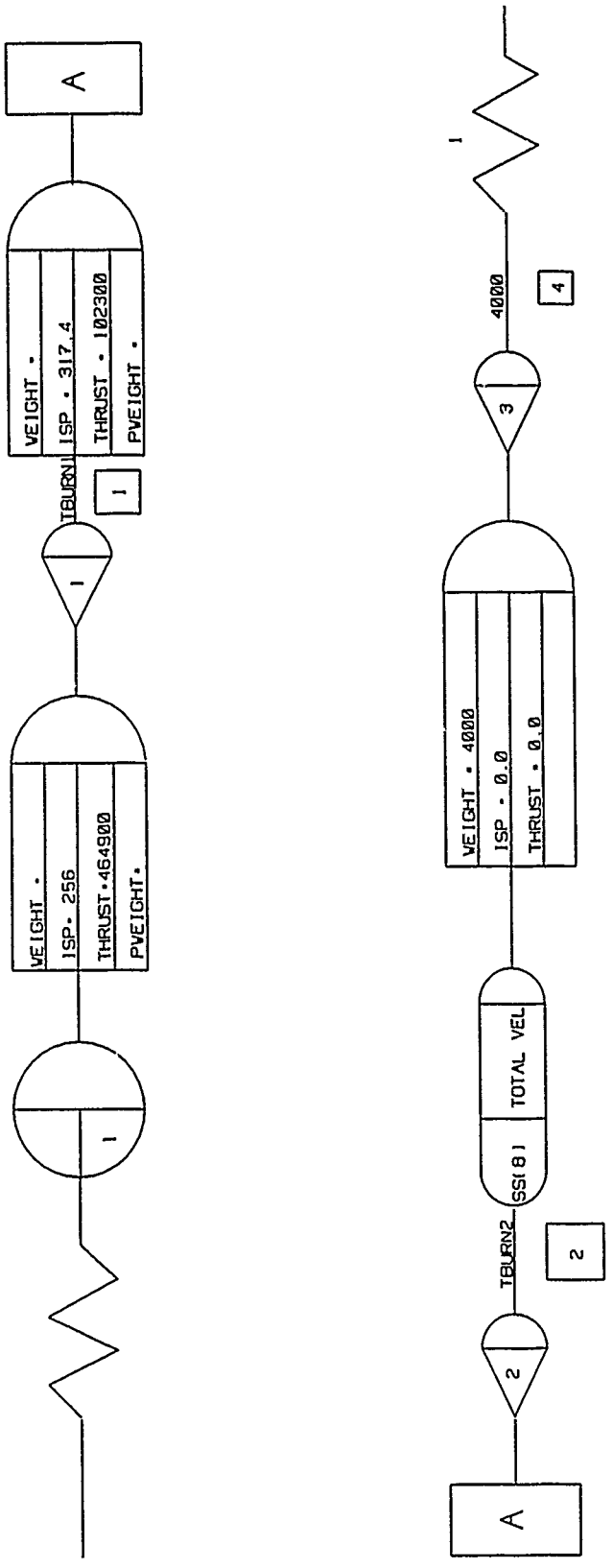


Figure A.1. Slam network diagram

3. Elevation above earth
4. X velocity
5. Y velocity
6. True velocity
7. Acceleration

A.3 BOND GRAPH MODEL

The SLAM model can not be linked directly to the bond graph model of the system. So, to provide missile dynamics inputs to the system for evaluation, a bond graph model of the missile dynamics was developed. The bond graph of the missile dynamics is shown in Figure A.2. The missile was modeled as a unit mass, in two dimensions, driven by two acceleration sources. Source SETT is acceleration due to thrust and SEGT is acceleration due to gravity. The missile bond graph is used to model only the position and velocities of the missile. Because of this, and the changing mass, sources represent acceleration and not force. Power is not modeled. SE1 is a function of the mass of the missile and the stage burn time. It changes discretely at stage separation and was modeled as:

$$SE1 = \frac{THRUST}{TOTALM - MFR \cdot TIME_s} \tag{A.15}$$

During stage one ignition TOTALM was modeled as:

$$TOTALM = MST1 + MST2 + MRV \tag{A.16}$$

During stage two ignition TOTALM was modeled as:

$$TOTALM = MST2 + MRV \tag{A.17}$$

During freefall, SE1 is set to zero.

where,

TOTALM = Total mass of missile

MST1 = Mass of stage one

MST2 = Mass of stage two

MRV = Mass of reentry vehicle

MFR = Stage mass flow rate

$TIME_s$ = Elapsed time of stage ignition

The gravity source SE2 is a function of position and was modeled as:

$$SE2 = \frac{\mu_e}{X^2 + Y^2} \tag{A.18}$$

Modulated transformers MTF1 and MTF2 are functions of position and were modeled as:

$$e_5 = e_6 \cdot \frac{X}{\sqrt{X^2 + Y^2}} \tag{A.19}$$

$$e_{10} = e_{11} \cdot \frac{Y}{\sqrt{X^2 + Y^2}} \tag{A.20}$$

$$e_{15} = e_{16} \cdot \frac{Z}{\sqrt{X^2 + Y^2}} \tag{A.21}$$

Modulated transformers MTF3 and MTF4 are functions of velocity and were modeled as:

$$e_3 = e_2 \cdot \frac{V_x}{\sqrt{V_x^2 + V_y^2}} \tag{A.22}$$

$$e_8 = e_7 \cdot \frac{V_y}{\sqrt{V_x^2 + V_y^2}} \tag{A.23}$$

$$e_{13} = e_{12} \cdot \frac{V_z}{\sqrt{V_x^2 + V_y^2}} \tag{A.24}$$

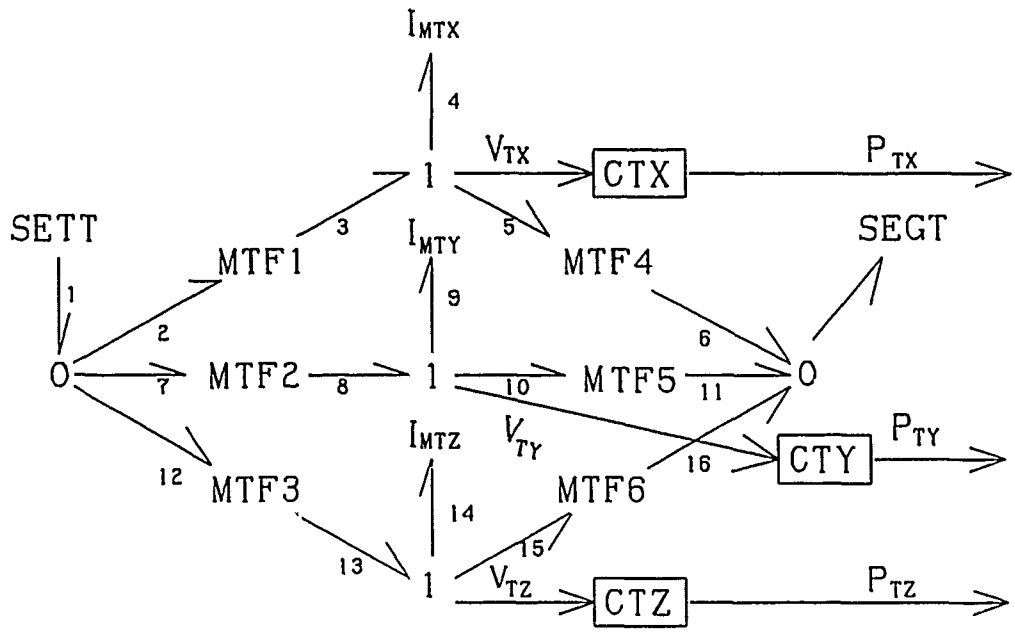


Figure A.2. Missile dynamics bond graph

The initial conditions of the integrators were set to the X and Y position of Kansk USSR. The output variables that are readily available from the model are:

1. X position
2. Y position
3. X velocity
4. Y velocity

The bond graph model was verified by checking the missile trajectory, missile velocity at second stage burnout, and missile position on impact.

Appendix B. *REFERENCE FRAMES AND ROTATIONAL TRANSFORMATIONS*

B.1 TRANSLATION OF TARGET VECTOR FROM STRUCTURE FRAME TO MIRROR FRAME

This appendix will describe the method by which the vector from the structure center of mass (SCOM) to the target expressed in the SCOM frame is converted to a vector from the primary mirror to the target expressed in the primary mirror frame. The method consists of two steps: a vector translation followed by a rotation. The math representation of this operation is (see Figure B.1):

$$\overrightarrow{mt} = {}^{m}A_{s} \cdot (\overrightarrow{ms} + \overrightarrow{st}) \tag{B.1}$$

$$\overrightarrow{ms} = -\overrightarrow{sm} \tag{B.2}$$

where,

$\overrightarrow{mt}$ = the position of the target with respect to the mirror

$\overrightarrow{st}$ = the position of the target with respect to the SCOM

$\overrightarrow{sm}$ = the position of the mirror with respect to the SCOM

$\overrightarrow{ms}$ = the position of the SCOM with respect to the mirror

${}^{m}A_{s}$ = rotation matrix from the SCOM frame to the primary mirror frame.

To understand the composition of the vector $\overrightarrow{sm}$, we must review the construction of the NASTRAN model that provides the vibrations of the structure. The nodes that make up the structure beams vibrate with respect to the SCOM. The primary mirror itself is a rigid disk attached to the structure at three points (or nodes) A,B,C. Due to constraints upon the rigid motion of the mirror, B and C positions and orientations depend solely upon point A's position and orientation. The modal analysis model provides the three translations and rotations of point A.

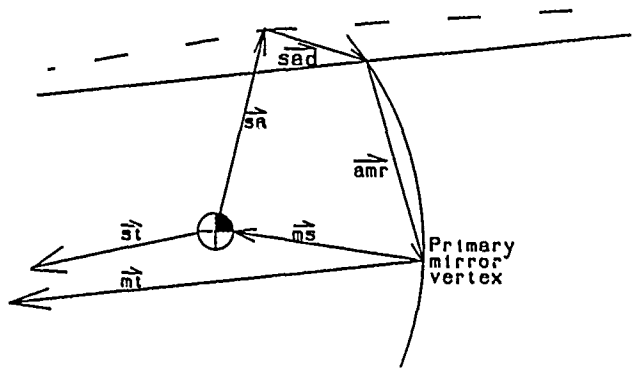


Figure B.1. Target vector in mirror frame

Once these values are found, then we can determine the position of any point on the mirror. Since the mirror is modeled as a rigid body, the angular orientation of any mirror point is equal to the angular orientation of the connection point A.

The vector $\overrightarrow{sm}$ consists of three separate parts. To find $\overrightarrow{sm}$, we must perform the following addition:

$$\overrightarrow{sm}=\overrightarrow{sa}+\overrightarrow{sad}+\overrightarrow{amr} \tag{B.3}$$

where,

$\overrightarrow{sa}$ = the position of point A without vibrations

$\overrightarrow{sad}$ = the position deviations of point A due to vibrations

$\overrightarrow{amr}$ = the position of the mirror vertex from the point A

Each vector is expressed in the SCOM frame so that addition to $\overrightarrow{st}$ is possible. $\overrightarrow{sa}$ is a constant vector that depends upon the system design. $\overrightarrow{sad}$ is calculated within the modal analysis model. $\overrightarrow{amr}$ is calculated via the equation:

$$\overrightarrow{amr}={}^{s}A_{m}\cdot\overrightarrow{am} \tag{B.4}$$

where,

$\overrightarrow{am}$ = the position of the mirror vertex with respect to the point A in the mirror frame. $\overrightarrow{am}$ is a constant vector that is defined by the design of the primary mirror.

${}^{s}A_{m}$ = rotation matrix from the mirror frame to the SCOM. ${}^{s}A_{m}$ is a changing rotation matrix that is calculated from the angular rotation of the connection point A. See Appendix C for a description of the calculation to find ${}^{s}A_{m}$. After $\overrightarrow{ms}$ is calculated, we add it to $\overrightarrow{st}$ to get the position of the target with respect to the mirror in the SCOM frame. The only calculation left to perform is the rotation of the vector from SCOM frame to the mirror frame by multiplying by the matrix ${}^{m}A_{s}$. ${}^{m}A_{s}$ is the inverse of ${}^{s}A_{m}$ and is found in a manner similar to ${}^{s}A_{m}$.

Appendix C. *RIGID BODY DYNAMICS - TRANSFORMATION BETWEEN FRAMES*

A three-axis system of attitude angles is used to describe the orientation or attitude of one body or reference frame relative to another. Specifically, a 3-2-1 sequence, involving all three distinctly indexed axes was used. In this appendix, first the transformation matrix is developed and then the transformation is implemented using bond graphs.

C.1 3-2-1 TRANSFORMATION BETWEEN FRAMES

To develop the transformation, start with a set of orthogonal unit vectors (x_0, y_0, z_0) aligned with the inertial frame (0). Any orientation of this set of vectors, relative to the inertial frame, can be obtained through the following rotation sequence:

1. Rotate the rigid body an angle ψ about the z_0 axis. This new position (frame 1) of the unit vectors is labeled

x_1, y_1, z_1.

2. Rotate the rigid body an angle θ about the new y_1 axis. This new position (frame 2) of the unit vectors is labeled x_2, y_2, z_2.

3. Rotate the rigid body an angle ϕ about the new x_2 axis. This final position (frame 3) of the unit vectors is labeled x_3, y_3, z_3.

The transformation matrix that rotates a vector $\vec{V}_1$ in frame 1 back into the inertial frame (0) is given by:

$$\vec{V}_0 = {}^0A_1 \vec{V}_1 \tag{C.1}$$

where:

$$ {}^{0}A_1 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.2} $$

The transformation matrix that rotates a vector $\vec{V}_2$ in frame 2 back into frame 1 is given by:

$$ \vec{V}_1 = {}^{1}A_2 \vec{V}_2 \tag{C.3} $$

where:

$$ {}^{1}A_2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{C.4} $$

The transformation matrix that rotates a vector $\vec{V}_3$ in frame 3 back into frame 2 is given by:

$$ \vec{V}_2 = {}^{2}A_3 \vec{V}_3 \tag{C.5} $$

where:

$$ {}^{2}A_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{C.6} $$

then,

$$ {}^1A_3 = {}^1A_2 \ {}^2A_3 = \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{C.7} $$

and,

$$ {}^0A_3 = {}^0A_1 \ {}^1A_2 \ {}^2A_3 = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\cos\phi - \sin\psi\cos\theta & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{C.8} $$

C.2 BOND GRAPH IMPLEMENTATION

By defining velocity vectors $\vec{V}_0$ and $\vec{\omega}_0$ in the inertial frame where:

$$ \vec{V}_0 = \begin{bmatrix} \dot{X}_0 \\ \dot{Y}_0 \\ \dot{Z}_0 \end{bmatrix} \tag{C.9} $$

$$ \vec{\omega}_0 = \begin{bmatrix} \omega_{X0} \\ \omega_{Y0} \\ \omega_{Z0} \end{bmatrix} \tag{C.10} $$

and velocity vectors $\vec{V}_3$ and $\vec{\omega}_3$ in frame 3 where:

$$ \vec{V}_3 = \begin{bmatrix} \dot{X}_3 \\ \dot{Y}_3 \\ \dot{Z}_3 \end{bmatrix} \tag{C.11} $$

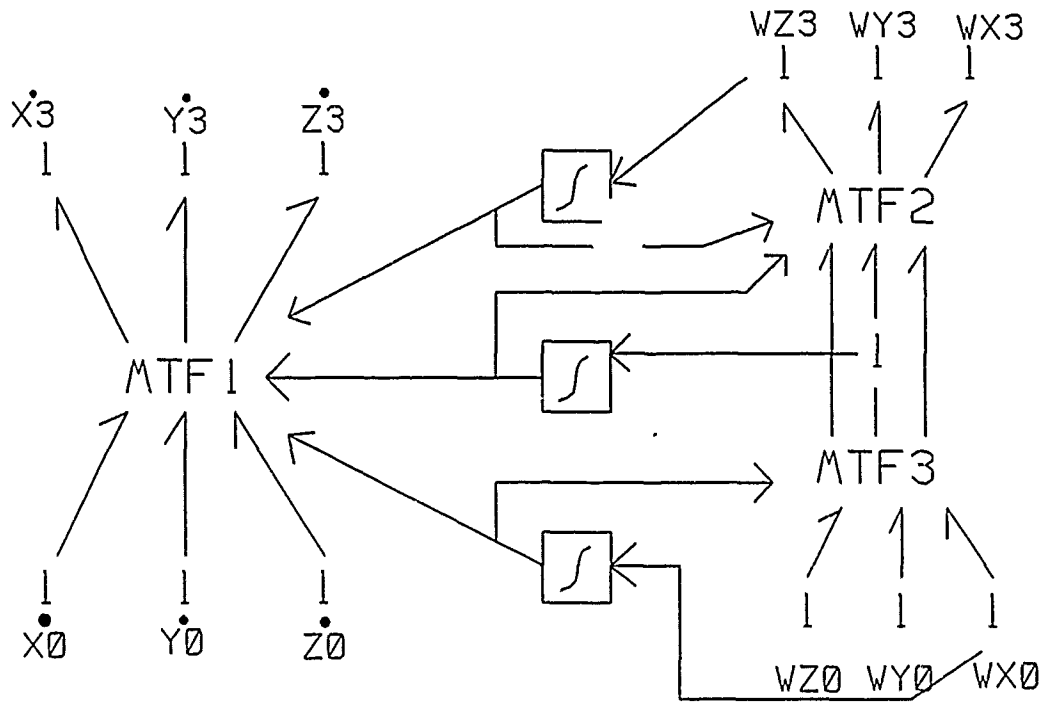


Figure C.1. 3-2-1 transformation bond graph

$$\vec{\omega_3} = \begin{bmatrix} \omega_{X3} \\ \omega_{Y3} \\ \omega_{Z3} \end{bmatrix} \tag{C.12}$$

A bond graph that transforms the velocity vectors from frame 3 to the inertial frame can be implemented as shown in Figure C.1.

The modulus for MTF1 is:

$$\bar{f}_{out} = {}^{0}A_3 \bar{f}_{in} \tag{C.13}$$

The modulus for MTF2 is:

$$\bar{f}_{out} = {}^{1}A_3 \bar{f}_{in} \tag{C.14}$$

The modulus for MTF3 is:

$$\bar{f}_{out} = {}^{0}A_1 \bar{f}_{in} \tag{C.15}$$

Appendix D. *CONTROLS*

The information provided in this appendix is intended to be an extension of the information provided in the chapter. Although all the information presented in the chapter is repeated here, the equations and data used in the chapter are developed in this appendix.

The satellite was designed to be moveable with the motion created by a controlling algorithm utilizing desired position information. Momentum wheels provide the torque input to the system to cause motion in a desired direction. The desired position information can come in three different forms.

The first form is a desired zero position, the position the satellite takes when no target information is available. The second form is target position as provided by another system. This information is passed to the system from another system that has detected launch and is handing off the tracking responsibilities. The third form is the optical position information. This is the position of the target as reported by the optics system.

The optical position information is the most crucial to the systems success. This is the only mode by which the satellite can provide any useful information to personnel on the ground or other tracking systems. As will be seen later, control via the optical information provides unique problems with nontrivial solutions.

D.1 BOND GRAPH

The basic bond graph of a single momentum wheel is shown in Figure D.1. A momentum wheel uses a high rotational momentum wheel mounted on a bearing and axle which is rigidly attached to the structure. A torque motor mounted on the axle provides a torque input to the wheel which in turn reacts and the input torque is then reflected into the structure thus causing an induced motion. In other words,

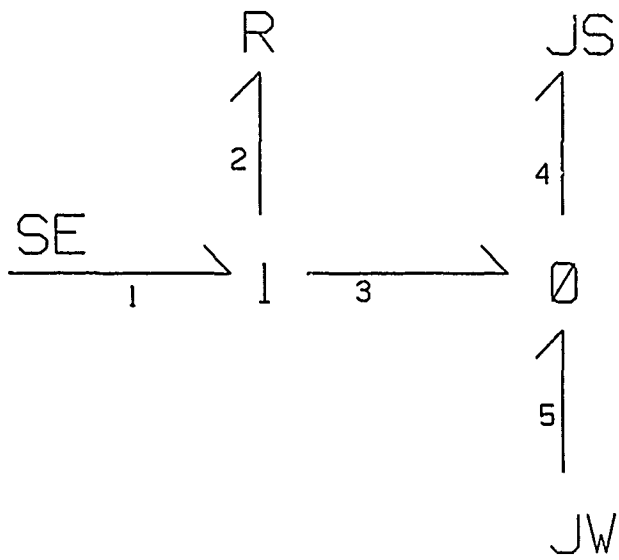


Figure D.1. Momentum wheel bond graph

the structure acts as the base from which the torque motor uses to push against to turn the momentum wheel. The associated bond graph shows a torque loss due to damping between the wheel and the structure (R) and even distributions of the net torque on the structure inertia (JS) and the wheel inertia (JW). As the bond graph indicates, the induced velocities in the wheel and the structure are in opposite directions.

As shown, the bond graph represents a two state system with the inertial momentums of the wheel and structure being the state variables. The bond graph provides a simple method for representing the physical dynamics of the momentum wheel structure.

Although the equations for this system could certainly have been determined using other techniques, bond graphs utilize energy transfer relationships to determine the dynamical equations of the system. This is similar to Newton-Euler formulations but provides a graphical method for showing the energy relationships within the system. This graphical representation can provide unique insights into a system and shows exactly how different systems interface together and energy is transferred

between them.

The determination of the state equations involves algebraic manipulation of the bond graph equations. Since the nodes (1 or 0) define the relationships between different sections of the bond graph, the majority of the algebraic manipulations take place at the nodes. For instance, the efforts present at a 1 junction sum algebraically to zero and the flows are all the same. Conversely, the efforts present at a 0 junction are all equal and the flows sum algebraically to zero. With this in mind, the state equations of the system can be determined.

At the 1 node,

$$f_1 = f_2 = f_3 \tag{D.1}$$

$$E_1 = SE = E_2 + E_3 \tag{D.2}$$

and at the 0 node,

$$f_4 = f_3 + f_5 \tag{D.3}$$

$$E_3 = E_4 = E_5 \tag{D.4}$$

The constitutive laws for the inertial elements are:

$$f_4 = \frac{p_4}{JS} \tag{D.5}$$

$$f_5 = \frac{p_5}{JW} \tag{D.6}$$

where,

$$\dot{p}_4 = E_4 \tag{D.7}$$

$$\dot{p}_5 = E_5 \tag{D.8}$$

The algebraic manipulations can now be performed on these equations. First use Equations D.7, D.4, and D.2 to formulate:

$$\dot{p}_4 = E_4 = E_3 = SE - E_2 \tag{D.9}$$

Equation D.9 is entirely defined as a state equation except for the E_2 term. However,

$$E_2 = f_2 \cdot R \tag{D.10}$$

so by using Equations D.1, D.3, D.5, and D.6:

$$f_2 = f_3 = f_4 - f_5 = \frac{p_4}{JS} - \frac{p_5}{JW} \tag{D.11}$$

then Equation D.9 becomes:

$$\dot{p}_4 = SE - R(\frac{p_4}{JS} - \frac{p_5}{JW}) \tag{D.12}$$

Finally since $E_4 = E_5$, the state equations for the bond graph and thus the system are:

$$\begin{bmatrix} \dot{p}_4 \\ \dot{p}_5 \end{bmatrix} = \begin{bmatrix} -\frac{R}{JS} & \frac{R}{JW} \\ -\frac{R}{JS} & \frac{R}{JW} \end{bmatrix} \begin{bmatrix} p_4 \\ p_5 \end{bmatrix} + SE \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{D.13}$$

$$\begin{bmatrix} p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{JS} & \frac{-1}{JW} \end{bmatrix} \begin{bmatrix} p_4 \\ p_5 \end{bmatrix} \tag{D.14}$$

It is now possible to determine the transfer function of this system using the state equation matrices. The characteristic polynomial of this system is:

$$| sI - A | = s^2 + s(\frac{R}{JS} - \frac{R}{JW}) \tag{D.15}$$

and thus the resolvent matrix is:

$$\phi(s) = \frac{1}{s^2 + s(\frac{R}{JS} - \frac{R}{JW})} \begin{bmatrix} s - \frac{R}{JW} & \frac{R}{JW} \\ \frac{-R}{JS} & s + \frac{R}{JS} \end{bmatrix} \tag{D.16}$$

The transfer functions of this system are given as the transformation of the input signal, SE, to the outputs, p_4 and p_5. In matrix terms these are given by:

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{JS} & \frac{-1}{JW} \end{bmatrix} \frac{1}{s^2 + s(\frac{R}{JS} - \frac{R}{JW})} \begin{bmatrix} s - \frac{R}{JW} & \frac{R}{JW} \\ \frac{-R}{JS} & s + \frac{R}{JS} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{D.17}$$

where G_1 represents the transfer function from the input torque to the rigid body rotational velocity. This transfer function will be referred to as the forward transfer function, GTF.

$$GTF = \frac{\frac{1}{J_s}}{S + \frac{R(JW - JS)}{JW \cdot JS}} \tag{D.18}$$

This represents an inherently unstable system that must be controlled through feedback loops and compensators to provide desired time response characteristics. The instability is readily seen as being a pole of the transfer function existing in the right half of the s-plane. This pole results because the inertial properties of the structure are likely to be greater than those of the momentum wheel.

Further investigation of the physical system reveals the damping factor, R, is considered to be so small as to be negligible. Since R is the coefficient of friction of the bearing between the momentum wheel and the structure, the assumption that the value of R is very small should be valid. As such, the pole in the right half plane

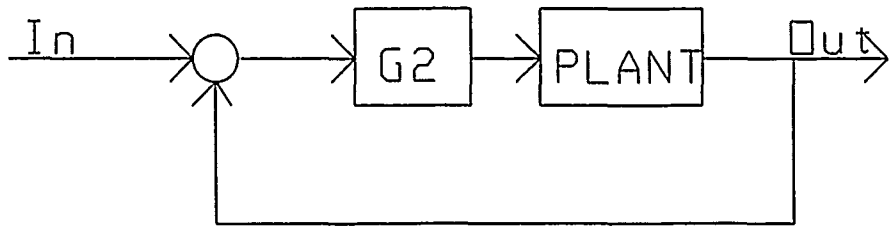


Figure D.2. Attitude controller block diagram

of Equation D.18 is nearly at the origin. To simplify the model and reduce the number of states, the damping is considered to be zero and eliminates the need for incorporating the momentum wheel inertia in the controls bond graph. The transfer function is then reduced to a simple integrator with the inverse of the structure inertia as the gain component:

$$GTF = \frac{\frac{1}{JS}}{S} \tag{D.19}$$

The intent of the design is to provide a control system that produced the desired structural rotations in a minimum of time without creating excessive vibrations in the structure. Therefore, the design was based on developing a compensator to give a particular overshoot and a particular settling time. The attitude control system block diagram is shown in Figure D.2. This figure shows the compensator G2 and the plant of Equation D.19. Different settling times were tried during the optimization portion of the research while keeping the overshoot the same.

The compensator design is based on root locus techniques. As seen in Equation D.19, the open loop transfer function of the plant has two poles at the origin when an integrator is added. The integrator is used to provide position information instead of velocity. The desired roots are found in the following fashion. First pick a desired overshoot and a desired settling time. Given the magnitude of the system response the first time it overshoots the final value is M_p, then the system parameters to give this value are:

$$M_p = 1 + \exp \frac{-\zeta\pi}{\sqrt{1-\zeta^2}} \tag{D.20}$$

Solving for ζ gives the desired damping ratio of the final closed loop system based on a simple second order response. With ζ known and given a desired settling time, the desired natural frequency of the second order response can be found via:

$$T_s = \frac{4}{\zeta\omega_n} \tag{D.21}$$

Now the desired roots of the characteristic equation are given by:

$$\sigma = \omega_n\zeta \tag{D.22}$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \tag{D.23}$$

where,

ω_d = desired root imaginary part

σ = desired root real part

The final trick in this derivation is to select compensator poles and zeros to give the desired effect. In reality, there are an infinite number of choices that will yield the desired roots. However, the idea is to select roots that not only yield the desired closed loop roots but also make the other roots of the characteristic equation non-dominant. A root is considered non- dominant if the real part of the root is at least six times to the left of the real part of the dominant root. If not all roots besides the desired roots are non-dominant, a simple second order response is not achieved. It is then up to the design engineer to either move the compensator roots or decide if the resulting time response characteristics are satisfactory.

The choice of the compensator poles and zeros can be illustrated as shown in Figure D.3. Given the desired roots, $\sigma_d \pm j\omega_d$, the sum of the angles contributed by the open loop poles and zeros must sum to 180°. From Figure D.3:

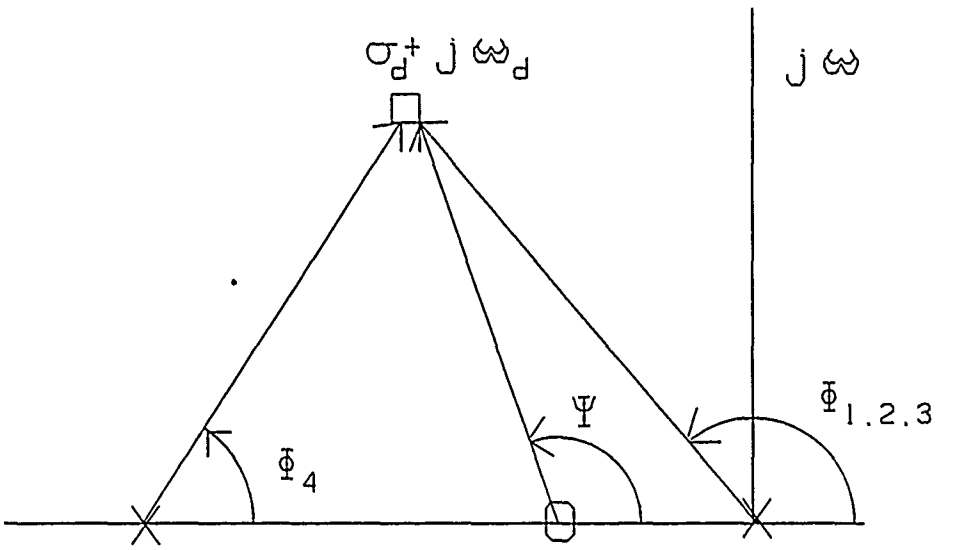


Figure D.3. Root locus angle conditions

$$\phi_{1,2,3} + \phi_4 - \psi_1 = 180° \tag{D.24}$$

where ϕ_4 is the angular contribution given by the compensator pole and ψ_1 is the angular contribution from the compensator zero. Care must be taken in choosing either the zero or the pole to ensure that closed poles do not appear in dominant positions. However, if dominant poles or zeros are present, it may be possible to place one of the compensator poles or zeros over the dominant roots and effectively cancel their effects. As will be seen, the choice of the compensator poles and zeros for this system were further complicated.

D.2 POLE-ZERO SELECTIONS

As mentioned in the text, the resulting system had to be modified to reflect a type 3 system in order to effectively track an accelerating missile. This in fact resulted in the need for placing a pole of the compensator at the origin. A quick application of the angle conditions reveals that the compensator must contain at

least 2 zeros in order to obtain the desired roots. The resulting compensator would be a second order over first order compensator. The problem with this compensator is that it is not physically realizable. Therefore, a second compensator pole must be added resulting in a second order over second order compensator.

Theoretically, the second compensator pole could be placed as far to the left as needed. However, physically realizable compensators limit this. Unfortunately, that was not the biggest problem. By looking at the angle conditions for this system, it becomes evident that the angular contributions of the two zeros must be large. So large in fact that they will result in real poles existing in positions more dominant than the desired roots. The solution to this problem would be to increase the order of the compensator to allow for a better selection of the poles and zeros. This was disallowed for the system model because the simulation software was limited in the number of allowable state variables and the limit had been reached. Therefore, a compromise was reached.

A pole was added at the origin to increase the system type from 2 to 3. However, a zero was added very close to the origin. This, in effect, canceled the effects of the pole at the origin. The system still exhibited type 2 behavior, but tracking improvement was seen. The tracking improvement was not a result of the system being a type 3, but was the result of the pole zero combination near the origin appearing as a lag compensator. This has the effects of raising the percent overshoot and increasing the settling time. However, it also has the effects of increasing the error coefficient, effectively lowering the tracking error.

The selection of the remaining pole and zero were accomplished by placing the zero at the point on the real axis corresponding to the real part of the desired dominant roots. This ensures that the resulting roots from this branch of the root locus will not be more dominant than the desired roots. The final pole of the compensator is chosen to satisfy the angle conditions.

Table D.1. Compensator designs.

Settling Time	Desired Roots	Compensator
1.0 sec	$-4 \pm j8.618$	Gain $\cdot \frac{(s+4)(s+.05)}{s(s+14.297)}$
1.44 sec	$-2.778 \pm j7.001$	Gain $\cdot \frac{(s+2.778)(s+.05)}{s(s+9.446)}$
2.5 sec	$-1.6 \pm j4.033$	Gain $\cdot \frac{(s+1.6)(s+.05)}{s(s+5.473)}$
3.56 sec	$-1.124 \pm j2.832$	Gain $\cdot \frac{(s+1.124)(s+.05)}{s(s+3.867)}$
4.0 sec	$-1 \pm j2.15442$	Gain $\cdot \frac{(s+1)(s+.05)}{s(s+3.651)}$

D.3 COMPENSATOR DESIGNS

Five compensators were designed for this research. Each of these were designed for the optimization portion of the research to study the effects of controller designs on the system performance. Each compensator was designed for an overshoot of 1.2 and a particular settling time. The overshoot was maintained as constant for each design while the settling time was varied. The settling time became a design variable for the optimization study. Table D.1 gives the compensator designs as well as the settling time and desired dominant roots.

Appendix E. *MODELING THE PHYSICAL SYSTEM*

E.1 ORIENTATION

The data and assumptions behind modeling the system depend on the input restrictions to SDRC Ideas, MSC/NASTRAN, and our design simplifications.

The system is composed of flexible subsystems and rigid subsystems. This categorization deals only with how we describe the physicality of the subsystems; it has no bearing on the function of the subsystem. The only flexible subsystem is the assembly of beams which define the tetrahedron frame. The momentum wheels and the optical system are modeled as rigid bodies.

The tetrahedron is oriented with the origin at the center of gravity (CG). The cartesian coordinate system has the XZ plane containing one of the beams. The tetrahedron apex is the frame joint which is behind (closest to) the detector plate. The optical axis defines the z-axis of the entire system, with the negative direction being towards the apex. All of the beams are considered to be joined by a weld at their ends. More specifically, these joints have no rotational degrees of freedom (DOF) relative to the beams, and also the corresponding translational DOF are fixed. The entire system, however, is dynamically in free-body motion.

The beams are modeled by finite element theory. The elements are linear Timeshinko beam elements. The frame is built in the SDRC Ideas Engineering Analysis module under the orientation discussed above. These beams are of uniform cross-sections and densities. The cross-section considered is solid circular. The material considered is a generic form of solid graphite/epoxy.

The optical components are considered to be rigid bodies; however, Ideas does not model rigid bodies with solid elements. An equivalent modeling scheme defines the center of gravity and attachment points of the optical system with a massless rigid element. A lumped mass element is defined on the CG node and provides

the required mass and inertia matrices. In the initial design phase, the optical components are all rigid with respect to each other. This aspect is modeled as a single lumped mass representing the entire optical system. Future design iterations require that the motion of the individual components be independent with respect to each other. This modeling requirement replaces the optical lumped mass with multiple lumped masses. These lumped masses are defined for each optical component and are connected by beam elements attached directly to nodes on the frame.

The momentum wheels are modeled with lumped mass elements; this procedure is the same as for the optical components. However, the lumped masses are connected directly to each of the base beams' center node. Great care is needed to insure that the body frames (coordinate systems) of each of the momentum wheels are properly aligned in their individual orientations. The system's global body frame is not aligned with any of the momentum wheel frames.

In Ideas, the system is constructed with an arbitrary origin. After its assembly, the CG is determined and the global origin is placed there. The NASTRAN bulk data deck is created by converting the Ideas model using the Ideas interfacing commands. The NASTRAN executive and control decks are added, and the modal analysis is run to produce the eigenvectors and modal matrices. These procedures are repeated for changes in the physical systems' design parameters.

E.2 MOMENTUM WHEELS

There are three momentum wheels which comprise the control system, each of the same mass and shape. The total mass is determined by (Agrawal, B. N., 1986):

$$MAC = 65 + 0.022(MSC - 700) \tag{E.1}$$

where MAC = mass of control system and MSC = mass of total system. If MMW = mass of an individual momentum wheel and MS = mass of the system without

MAC, then:

$$3MMW = 65 + 0.022(3MMW + MS - 700) \tag{E.2}$$

and so

$$MMW = 16.985 + .075MS \tag{E.3}$$

Therefore, the masses of the optical system and the frame structure are determined; then the individual momentum wheel masses are found.

The shape of the wheels was arbitrarily set as disks or thick cylinders. Since the momentum wheels mass can include an exterior casing, a motor, electronics, etc., the density was set at less than half that of solid aluminum to 1000 kg/m^3. Therefore, the volume of each wheel was found by dividing MMW by 1000. By defining a thick cylinder as having a radius which is four times larger than its thickness, the radius is produced by:

$$r = \left[\frac{4 \cdot \text{wheel volume}}{\pi}\right]^{\frac{1}{3}} \tag{E.4}$$

The densities and dimensions of the wheels are then used to find their inertias relative to the principle axes.

E.3 OPTICAL SYSTEM

The shape of each optical component is approximated as a thick cylinder. Note that this shape is only an approximation for the optical components inertial properties; the optical ray tracing algorithm does not concern itself with these approximations. The construction of the primary and secondary mirrors are assumed as honeycomb structures, manufactured out of a ceramic, ULE (Shannon and Wyant, 1979). The density for ULE is 2.20E3 kg/m^3. The effective density of these components is obtained from:

$$\rho_{eff} = \rho_{solid}(2 \cdot \alpha + 2 \cdot t) \tag{E.5}$$

With $t = 0.10$ m , and $\alpha = 0.05$ m (which is a very conservative assumption), the effective density is 660 kg/m^3.

The total thickness of the primary mirror and the secondary mirror are assumed to be twice the vertical distance measured from the vertex to the rim. This distance is expanded from the optical definitions as:

$$Z = \frac{2r}{4f} \tag{E.6}$$

where r and f are the radius and the focal length, respectively. The lense thickness is derived in the same manner, but the density is simply that of an optical-quality glass. The imaging detector is of little mass and is neglected. However, this detector is mounted on a detector plate, which is considered to be a rigid disk. This disk is arbitrarily 10 cm in diameter and 50 mm thick, with a density of 2700 kg/m^3. This disk could have been completely ignored in the current simulation; however, it was included to allow for an expansion of the research in the future. This expansion would be the inclusion of motion between the individual optical components. If the detector plate were ommitted, the relative motion between the lense and the plate would be unattainable.

After all the optical components masses and intertias have been determined, they are combined into a single optical system mass and inertia tensor. This reduction is possible only for the first concept of the total system, with no relative motion between optical components, the inertia tensor will never change relative to the optical system's principle axes.

Appendix F. *EXACT RAY TRACE DESCRIPTION*

Typical optics systems involve complex geometrical configurations of geometrical shapes used to reflect or refract light rays. The problem with these systems lies in the fact that exact ray tracing can be very difficult for relatively simple systems. As such, a paraxial approach is typically used as an approximation of the actual performance. However, if a method can be found to simply implement an exact ray trace, a more accurate picture of the systems performance can be achieved. This appendix demonstrates how an exact ray trace method is implemented in the cassegrainian telescope optics system.

F.1 GEOMETRIC RAY TRACE

The basic premise behind the exact ray trace method is the translation of a given ray from one coordinate system (primary mirror) to another coordinate system (secondary mirror). To begin the ray trace, the orientation of the point in question with respect to the first coordinate system must be known. For the tracking system, all of this data is known beforehand as the exact position of the missile with the satellite is known at all times. To fully understand the geometry behind the exact ray trace, an example is given. To simplify the example, begin the ray trace inside the optical system and trace from the primary mirror to the secondary mirror. This method can be used anywhere in the optics system without loss of generality. Starting at the primary and tracing to the secondary is done only for convenience.

The first step is to attach a reference frame to the secondary mirror. The origin of the frame is attached to the vertex of the mirror with the positive z-axis pointing towards the primary mirror. The x and y axes are oriented tangent to the mirror surface and form a right handed coordinate system as shown in figure F.1. The incident ray to the secondary mirror is defined as having a given orientation to the reference frame attached to the secondary mirror. The vector $\vec{\rho}_{prev}$ defines

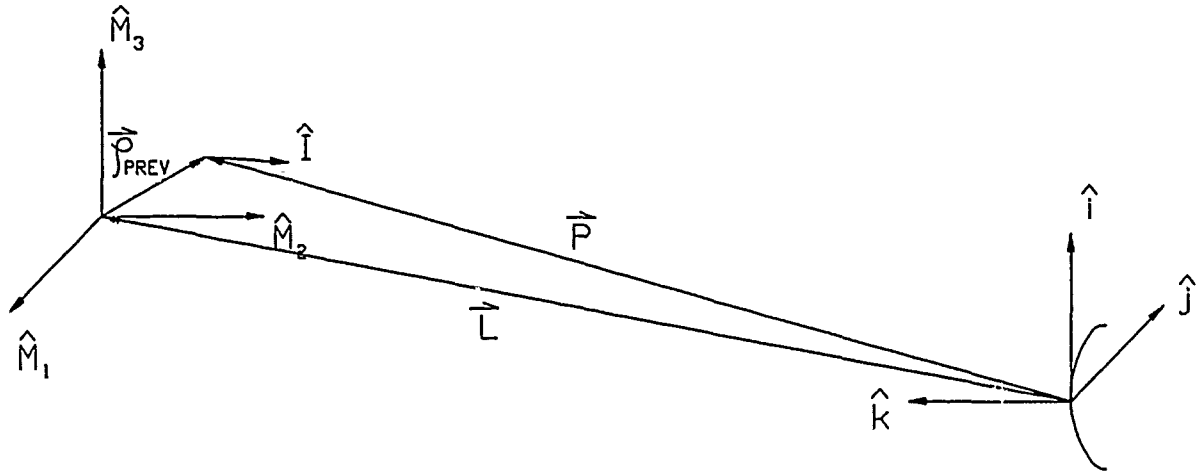


Figure F.1. Exact ray trace vector definitions

the location of the rays originating point on the primary mirror with respect to the primary mirror reference frame.

Now define a vector $\vec{L}$ that locates the origin of the primary mirror reference frame. The rays originating location can now be defined as:

$$\vec{L} + \vec{\rho}_{prev} = \vec{P}. \tag{F.1}$$

Note that coordinate transformations were used between reference frames.

In order to determine the striking point on the secondary mirror surface, it is convenient to first determine the striking point on the x-y plane and then determine the striking point on the mirror from the plane. Defining the distance from the rays originating point to the x-y plane L_1, and define the unit vector in the direction of the incident ray as shown in Figure F.2:

$$\hat{I} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{F.2}$$

then the addition of $\vec{P}$ and $L_1\hat{I}$ strikes the plane when the z component goes to zero. Defining

$$\vec{P} = \begin{bmatrix} d \\ e \\ f \end{bmatrix} \tag{F.3}$$

then,

$$L_1 c + f = 0 \tag{F.4}$$

or,

$$L_1 = \frac{-f}{c} \tag{F.5}$$

The trick now is to determine the distance from the incident point on the x-y plane to the mirror surface. For a spherical mirror, the surface is defined by vector ρ as

$$\vec{\rho} = \begin{bmatrix} x \\ y \\ z - r \end{bmatrix} \tag{F.6}$$

where,

$$x^2 + y^2 + z^2 = r^2 \tag{F.7}$$

Now calling the vector from the frame origin to the incident point on the x-y plane, $\vec{T}$,

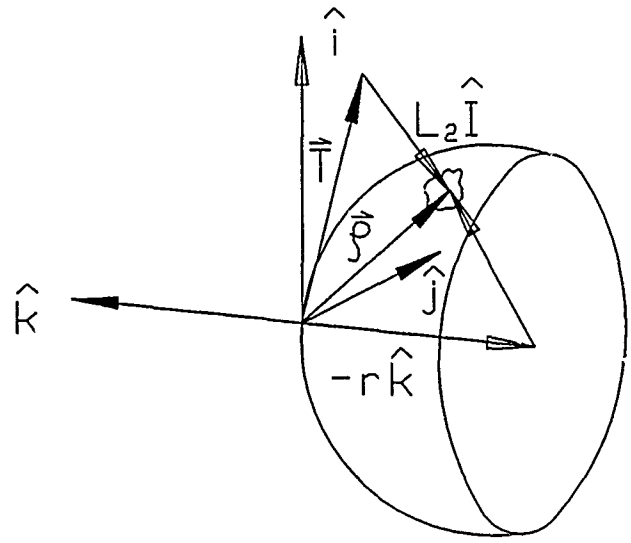


Figure F.2. Vector definitions for surface striking point.

$$\vec{T} = \begin{bmatrix} g \\ h \\ 0 \end{bmatrix} \tag{F.8}$$

and calling the distance from the plane to the striking point on the mirror L_2, results in

$$\vec{T} + L_2\hat{I} = \vec{\rho} \tag{F.9}$$

Multiplying this out gives:

$$(L_2a + g)\hat{i} + (L_2b + h)\hat{j} + (L_2c + r)\hat{k} = x\hat{i} + y\hat{j} + z\hat{k} \tag{F.10}$$

where the radius of the sphere has been moved to the left side of the equation. In order for these vectors to be equal and satisfy the solution to the mirror surface equation, Equation F.7, the following must hold:

$$(L_2a+g)^2+(L_2b+h)^2+(L_2c+r)^2=r^2 \qquad \text{(F.11)}$$

Using this expression, the value of L_2 can be determined, and thus $\vec{\rho}$. The reflection off of the mirror is easily defined in matrix notation as:

$$\hat{R}=\left(\mathbf{I}-2\hat{N}\hat{N}^T\right)\hat{I} \qquad \text{(F.12)}$$

where $\hat{N}$ is defined as the surface normal at the incident point on the mirror surface and $\hat{R}$ is the reflected ray. Once accomplished, $\vec{\rho}$ becomes $\vec{\rho}_{prev}$ for the next mirror, and $\hat{R}$ becomes $\hat{I}$.

F.2 EXACT RAY TRACE MATRIX NOTATION

This same approach can be extended to a matrix notation (Redding, D., 1989). The general matrix approach centers around one central matrix, the mirror matrix. This matrix is defined as:

$$\mathbf{M}\equiv\left(\mathbf{I}-e^2\hat{\psi}\hat{\psi}^T\right) \qquad \text{(F.13)}$$

where e is the eccentricity of the geometrical shape of the mirror and $\hat{\psi}$ is a unit vector pointing from the vertex of the mirror towards the focal point. The mirror matrix can be compared to Equation F.7 as it defines the surface of the mirror, at least in part In addition, this matrix allows for the principle axis of the conic shape to not lie on any of the principle axes of the coordinate system. This is explained further when flexible optics is discussed. With this definition, the surface of the mirror is defined by:

$$\vec{\rho}^T\mathbf{M}\vec{\rho}+2\vec{N}_o^T\vec{\rho}=0 \qquad \text{(F.14)}$$

where $\vec{N}_o$ is a vector along the surface normal and whose magnitude is equal to the

radius of curvature evaluated at the vertex. The vector $\vec{N}$ at an arbitrary point on the mirror surface is then:

$$\vec{N} = \vec{N}_o + \mathrm{M}\vec{\rho} \tag{F.15}$$

With these definitions and using the same vectors in the geometric derivation, the following solution can be achieved:

$$\left(\hat{I}^T \mathrm{M} \hat{I}\right) L^2 + 2\hat{I}^T \left(\mathrm{M}\hat{P} + \vec{N}_o\right) L + \hat{P}^T \left(\mathrm{M}\hat{P} + 2\vec{N}_o\right) = 0 \tag{F.16}$$

where L is a scaler equal to the total distance travelled by the incident ray from the previous surface. The reflected ray can then be calculated as was done in the geometric derivation.

The position of the missile with respect to the primary mirror is determined by the orientation of the system in inertial space and the position of the missile in inertial space. This vector acts as the incident ray, $\hat{I}$, to begin the ray trace. The system position is determined by the rigid body rotation model. Once the missile position is known, this position is provided as input to the exact ray trace software to begin a ray trace.

F.3 EXACT RAY TRACE SOFTWARE

The exact ray trace software utilizes the matrix approach described in the previous section. The software utilizes double precision in all calculations as most calculations involve small differences of large numbers. The ψ vectors of each component are contained in a data file that also contains all the geometric parameters of all the optical components. Also contained in this data file are the distance vectors, $\vec{L}$, that define the directed distances between components, the geometric focal lengths

of each component, the eccentricity of each component, and the index of refraction of the lens.

Before the ray trace is begun, the software utilizes information on the geometric shapes of each of the optical system components to set flags used throughout the ray trace. The first of these is a determination of the conic section of each of the components and the direction the light rays will approach from. In other words, the shape (i.e. spherical, elliptical, parabolic, or hyperbolic) is determined and whether it is concave or convex with respect to the incident ray. This determination is critical to calculating the length the incident ray travels from component to component.

The software also calculates the thickness of the lens based on the diameter of the lens, the eccentricity and focal lengths of both surfaces, and the orientation of the surfaces (concave or convex). With this calculation it is possible to quickly change the geometric properties of the lens and determine the best configuration. This calculation also allows for dismissal of thins lens approximations as the ray can be traced completely through the lens.

F.3.1 Incident Ray Calculations The incident ray to the optics system begins as the vector from the primary mirror vertex to the missile. However, this vector alone does not provide much useful information on the properties of the optical system. A much more useful approach is the use of wavefronts, many rays of incidence striking the entire surface of the primary mirror. It is the recombining of the wavefront on the image plate that provides the useful information on the inherent properties of the optical system.

Although a continuous wavefront was not practical for this software (in terms of calculation requirements), a method was devised to simulate a wavefront. This is based upon taking a several incident rays that would strike the primary mirror in a symmetric fashion. It was decided to trace 32 incident rays to simulate a wavefront. Each of these incident rays emanates from a single point source on the missile. The

incident rays are calculated such that they strike the primary mirror in an evenly distributed fashion.

The incident rays are created by first determining the desired strike points on the primary mirror. Each incident can then be determined by calculating the vector from the desired point on the mirror back to the originating point source. Once accomplished, 32 incident rays are available for ray tracing through the optics system.

F.3.2 Ray Trace on Mirrors The exact ray trace for reflecting surfaces is carried out in the manner described earlier. This section will give the details for determining the surface strike point and the reflected ray direction.

The first calculation to be carried out is the determination of the semi-latus rectum of the surface. This value is used to calculate the vector $\vec{N}_0$ as defined earlier. The value of the semi-latus rectum is found via:

$$p = f(1 + e) \tag{F.17}$$

where f is the focal length and e is the eccentricity of the surface. Note the convention for a flat surface is one of a spherical surface (eccentricity, e, of 0) with an infinite focal length, f.

Second, the mirror matrix of Equation F.13 is calculated. As mentioned, this matrix serves as the foundation for all subsequent determinations. In order to calculate this matrix, the ψ unit vectors must already be known. These vectors were defined as the system was designed, but the orientation of these vectors is arbitrary. These vectors simply define the direction of the principle axis of the mirror. Typically, these vectors share a common axis but may have 180° difference in direction. If one of these vectors does not lie on the optical axis, then that component has been rotated. This feature allows the optical components to be placed off axis

and also allows for sensitivity analysis due to vibrations or structural deformations between components.

The vector $\vec{N}_0$ is now calculated as defined earlier. Once obtained, the length the ray must travel to strike the mirror surface may be determined. Calling the vector locating the origination point of the incident ray $\vec{p}$, the distance the ray must travel is accomplished by solving Equation F.16 for L. This involves the solution to a quadratic equation. The question remaining is whether to use the plus or minus sign in the quadratic equation. As it turns out, the correct sign to use is dependent upon the type of conic section the surface represents and if it is concave or convex. This then refers back to the first part of the software where flags were set. Utilizing the proper logic, the software is able to determine if the plus or minus sign is needed based on the settings of the flags. Now all of the information is available to calculate the point on the mirror where the incident ray strikes. Calling the vector from the vertex to the striking point $\vec{\rho}$,

$$\vec{\rho} = \vec{p} + L\hat{i} \tag{F.18}$$

The vector $\vec{\rho}$ may not strike the mirror surface if the missile is near or beyond the field of view of the optical system. Therefore, a determination is made as to whether the magnitude of the vector $\vec{\rho}$ represents a point on the surface or not. If not, the ray trace for this particular ray is halted and the ray is assumed to be eliminated from the system. If the ray does strike the surface, the ray trace continues.

Now that the striking point on the surface is known, the reflected ray must be calculated as in Equation F.12. In order to accomplish this though, the surface normal $\hat{N}$ must be calculated. The vector $\vec{N}$ is calculated as in Equation F.15. The calculation of $\hat{N}$ is accomplished via

$$\hat{N} = -sign(\hat{i}^T \cdot \frac{\vec{N}}{N})\frac{\vec{N}}{N} \tag{F.19}$$

where $N = \mid \vec{N} \mid$. In this equation, the symbol $\cdot$ represents the dot product. Note

that $\frac{\vec{N}}{N}$ actually represents $\hat{N}$. However, this vector can take two directions. By using the formulation of Equation F.19, $\hat{N}$ is given the proper direction. The reflected ray may now be calculated using Equation F.12.

Thus the translation is complete, an incident ray was processed through a reflecting surface and became a reflected ray. This reflected ray now becomes the incident ray for the next surface. The process is repeated for every component in the system. It is also completed for the primary mirror hole although a reflected ray is not calculated. The primary mirror hole calculations are done only to determine if a given ray passes through the mirror hole. If not it is eliminated from the system. Also, reflected rays are not calculated at the lens, instead refracted rays are calculated.

F.3.3 Refracted Rays The exact ray trace calculations at a refractive surface is identical to those at a reflective surface. The only difference being the ray is refracted instead of reflected.

Recall the reflection of an incident ray at a point on a mirror surface was given in vector notation in Equation F.12. A similar notation is provided for a refracted ray. This equation is

$$\hat{r} = \frac{n_a}{n_b}\hat{i} - \frac{1 - \frac{n_a}{n_b}}{\sqrt{1 - (\frac{n_a}{n_b})^2 + (\frac{n_a}{n_b})^2(\hat{N}^T\hat{i})^2} - \frac{n_a}{n_b}\hat{N}^T\hat{i}}\hat{N} \qquad \text{(F.20)}$$

where, n_a is the index of refraction of the incident medium, n_b is the index of refraction of the refracted medium, and the other terms are as defined earlier.

The lens surfaces can be treated as two separate components in the system. The distance between the two surfaces was calculated early in the software and now can be used to translate a given ray from one surface of the lens to another. As the ray passes through each surface, the ray is refracted and becomes the incident ray for the next component.

F.4 CAPABILITIES

These calculations are exact calculations and do not contain any assumptions. Thus the ray trace reflects the actual performance of an ideal optical system (i.e. one with perfect mirrors, etc.). All aberrations present in the actual system will also be present in this exact ray trace formulation.

The software also collects statistical information on the image formed on the detector plate. Recall that the software traces a bundle of 32 rays emanating from a single point through the optical system. This software will calculate a centroid (mean) on the detector plate as well as the variance of the point spread. This is required as exact focussing is difficult to achieve for point sources off axis. The centroid acts as targeting information (i.e. where the missile is located) and the variance can act as a performance measure as to the quality of the 'spot' on the detector plate.

The software is also capable of forming an image on the detector plate. This is accomplished by tracing several bundles of rays through the system. Each bundle represents a given point on the object. If several points are taken off of the object, the image of the object can be reconstructed on the detector plate. This image will reflect the aberrations present in the system and can serve as another performance measure.

F.5 EXACT RAY TRACE SOFTWARE CODE

The code for the exact ray trace software is included. This code is the final version of the software used in the optimization of the satellite system. This code is based on matrix formulations developed by the Jet Propulsion Laboratory (Redding, D., 1989).

```
      PROGRAM MAIN
      IMPLICIT DOUBLE PREC       .A-H,O-Z)
      DIMENSION VPP(3),VPI      ,ROE(3)
CMNM       OPEN (7,FILE='MISSROE.DAT',STATUS='NEW')
      OPEN (2,FILE='DUMMY.DAT',STATUS='OLD')
CMNM       VPP1(1)=0.D0
CMNM       VPP1(2)=13.D6
CMNM       VPP1(3)=0.D0
CMNM       VPP(1)=VPP1(1)
CMNM       VPP(2)=VPP1(2)
CMNM       VPP(3)=VPP1(3)
      READ (2,*)JTIME
      READ (2,*)VPP(1),VPP(3),VPP(2)
      OPEN (3,FILE='VAR.DAT',STATUS='NEW')
      DO 176,I=2,360
      READ (2,*)JTIME
      READ (2,*)VPP(1),VPP(3),VPP(2)
      VPP(3)=-VPP(3)
      CALL SUPEROP(VPP,XBAR,ZBAR,JTIME,TA,POW,RMN,RV,PVR)
176   CONTINUE
      CLOSE (2)
      CLOSE (3)
C CALCULATE TIME-AVERAGED DATA
      TA = TA / 359.D0
      POW = POW / 359.D0
       POWINV = 1/POW
      RMN = RMN / 359.D0
      RV = RV / 359.D0
      PVR = PVR/359.D0
C WRITE DATA FOR RSM ANALYSIS
      OPEN (2,FILE='RSM.DAT',STATUS='NEW')
      WRITE(2,*)TA,RMN,RV,POW,POWINV,PVR
C      CLOSE (7)
      STOP
      END
      SUBROUTINE SUPEROP(VPP,XBAR,ZBAR,JTIME,TA,POW,RMN,RV,PVR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPP(3),VP(3),VPS(3),VSAT(3),VPSI1(3),VPSI2(3),X(32),
     1Z(32),ROE(3),XID(3,3),XMTX(3,3),XNO(3),XN(3),VI(3),XNH(3),
     2RM(3,3),R(3),XL1(3),XL2(3),XL3(3),VPSI3(3),VPSI4(3),VPL(3),
     3XL4(3),VPSI5(3),ARRAY(32,2)
C
C  ****************************************************************
C  *      VARIABLE DEFINITIONS
C  *      PMF -  PRIMARY MIRROR FOCAL LENGTH
C  *      PME -  PRIMARY MIRROR ECCENTRICITY
C  *      SMF -  SECONDARY MIRROR FOCAL LENGTH
C  *      SME -  SECONDARY MIRROR ECCENTRICITY
C  *      XL1 -  DIRECTED DISTANCE FROM PM TO SM
C  *      XL2 -  DIRECTED DISTANCE FROM SM TO LENS
C  *      XL3 -  DIRECTED DISTANCE FROM LENS TO IMAGE PLANE
C  *      DPM -  PRIMARY MIRROR DIAMETER
C  *      DPMH - APERTURE HOLE SIZE (DIAMETER)
C  *      ROE - VECTOR FROM VERTEX OF SURFACE TO THE
C  *            POINT OF INCIDENCE.
C  *      VPSI - VECTOR FROM VERTEX TO FOCUS
C  *      VPP - POSITION VECTOR OF LIGHT SOURCE IN
C  *            UNIVERSAL COORDINATES.
C  *      VP - SAME AS VPP ONLY IN MIRROR COORDINATES.
C  *      VI - UNIT VECTOR DEFINING INCIDENT RAY DIRECTION
C  *      XID - 3X3 IDENTITY MATRIX
C  *      XMTX - 3X3 MIRROR MATRIX
C  *      II - INCIDENT POINT COUNTER
C  *      XNO - NORMAL VECTOR OF MIRROR AT VERTEX WITH
C  *            MAGNITUDE EQUAL TO SEMI-LATUS RECTUM
C  *      SLR - SEMI LATUS RECTUM OF MIRROR
```



```
C  *      XN - NORMAL VECTOR OF SURFACE AT POINT OF INCIDENCE
C  *      XNH - UNIT VECTOR OF XN
C  *      RM - REFLECTION MATRIX OF MIRROR
C  *      R - UNIT VECTOR REPRESENTING DIRECTION OF REFLECTED RAY
C  *      PVR - POWER TO VARIANCE RATIO
C INPUT ALL OPTIC SYSTEM PARAMETERS
      CALL PARAMIN(VSAT,VPSI1,VPSI2,XL1,XL2,XL3,PMF,PME,
     1SMF,SME,XLF1,XLE1,XLF2,XLE2,DPM,DPMH,DSM,DL,VPSI3,VPSI4,
     2XNA,XNB,DIP,VPSI5,RES)
      NNN=0
C
C  ***** FIRST LETS CALCULATE SOME MAX MAGNITUDES FOR THE
C  ***** ROE VECTORS ON EACH OPTICAL ELEMENT
C
C   **  PRIMARY MIRROR .
      CALL MAGNITUDE(PMF,PME,DPM,PMM)
C   **  SECONDARY MIRROR
      CALL MAGNITUDE(SMF,SME,DSM,SMM)
C   **  APERTURE
      CALL MAGNITUDE(PMF,PME,DPMH,PMHM)
C   ** LENS
      CALL MAGNITUDE(XLF1,XLE1,DL,XLM)
C
C ***************************************************
C ***  THIS SMALL PART IS USED TO SET UP THE VARIABLES **
C ***  FOR THE FLAT SIDE OF A EQUICONVEX LENS.  IF THE **
C ***  LENS IS CHANGED, THIS SECTION MUST ALSO BE CHNGED*
C ***  FOR THIS VERSION, THE LENS HAS BEEN CHANGED TO ***
C ***  A COMPLEX LENS WITH BOTH SIDES BEING SPHERICAL ***
C ***************************************************
C ***************************************************
       XL4(1)=0.D0
       XL4(3)=0.D0
       PHI=90.D0-DACOSD(DL/(2.D0*XLF1))-DASIND(DL/(2.D0*XLF2))
       XL4(2)=XLF1-XLF2+((2.D0*XLF1*XLF2)/DL)*DSIND(PHI)
       CALL MAGNITUDE(XLF2,XLE2,DL,XLM2)
C
C ***************************************************
C ***************************************************
C GENERATE AN IDENTITY MATRIX
      CALL IDMATRX(XID)
C
C TRANSFER VPP TO PM FRAME - BECOMES VP
      CALL PMTRANS(VPP,VSAT,VP)
C CALCULATE XZ POINTS ON PM WHERE INCIDENCE OCCURS
C
      CALL XZCALC(DPM,DPMH,X,Z)
      DO 900 II=1,32
      FLAG=0.D0
C
C CALCULATE THE STUFF FOR THE PM
C
      CALL REFLECT(NCOMP,VP,VI,X,Z,II,XID,ROE,PMF,PME,VPSI1,R)
C
C NOW WE HAVE THE FIRST RELECTED RAY; IT BECOMES THE NEW POINT
C SOURCE FOR THE SM, VPS;
C
C TRANSLATE TO NEXT MIRROR
C
      CALL SMTRANS(XL1,ROE,R,VI,VPS)
C
C CALCULATE REFLECTION FOR SECOND COMPONENT
C
      NCOMP=2.D0
      CALL  COMPONENT2(NCOMP,VPS,VI,II,SME,SMF,VPSI2,XID,ROE,R,SMM,
     1FLAG)
```

```
C
C    *** CHECK TO SEE IF RAY IS STILL IN SYSTEM
      IF(FLAG.EQ.-1.D0) GO TO 900
C    *** OTHERWISE, CONTINUE WITH CALC
C
C   ***** MUST NOW DETERMINE IF RAY PASSES THORUGH APERTURE
      CALL LENSTRANS(XL1,ROE,R,VI,VPL)
      NCOMP=1.D0
      CALL COMPONENT4(NCOMP,VPL,VI,II,PME,PMF,VPSI1,XID,PMHM,
     1FLAG)
      IF(FLAG.EQ.-1.D0) GO TO 900
C    ********************************************************
C    ********    TRANSFORM GIVEN R VECTOR INTO LENS  *******
C    ********    FRAME BY CREATING THE NEW INCIDENT  *******
C    ********    RAY FROM R AND CALCULATE THE POSITION******
C    ********    VECTOR VPL THAT GIVES THE ORIGINATING******
C    ********    POINT OF THE RAY ON THE SECONDARY   *******
C    ********   MIRROR WRT VERTEX OF LENS CONCAVE SIDE ****
C    ********************************************************
      CALL LENSTRANS(XL2,ROE,R,VI,VPL)
C
C  ********************************************************
C  **********    NOW CALCULATE THE NECESSARY ROUTINES  ******
C  **********    TO GIVE THE NEW R VECTOR              ******
C  ********************************************************
      NCOMP=2.D0
      CALL COMPONENT3(NCOMP,VPL,VI,II,XLE1,XLF1  PSI3,XID,
     1ROE,R,XNA,XNB,FLAG,XLM)
      IF(FLAG.EQ.-1.D0) GO TO 900
      CALL LENSTRANS(XL4,ROE,R,VI,VPL)
      NCOMP=2.D0
      CALL COMPONENT3(NCOMP,VPL,VI,II,XLE2,XLF2,VPSI4,
     1XID,ROE,R,XNB,XNA,FLAG,XLM2)
      IF(FLAG.EQ.-1.D0) GO TO 900
      CALL LENSTRANS(XL3,ROE,R,VI,VPL)
      NCOMP=3.D0
      CALL COMPONENT2(NCOMP,VPL,VI,II,0.,1000000000.,VPSI,
     1XID,ROE,R,DIP,FLAG)
        WRITE (7,567)ROE(1),ROE(2),ROE(3)
567     FORMAT(1X,F17.10,F17.10,F17.10)
C
C    PUT NEW RAY INTO IMAGE PLANE ARRAY
C
      NNN=NNN+1
       ARRAY(NNN,1)=ROE(1)
       ARRAY(NNN,2)=ROE(3)
 900  CONTINUE
C
C    NOW CALCULATE MEAN VALUES FOR OUTPUT TO CONTROLLERS
C
      IF ( NNN.EQ. 0 ) THEN
      XBAR = 10000.D0
      RETURN
      ELSE
      ENDIF
      X1=0.D0
      XBAR=0.D0
      ZBAR=0.D0
      DO 124,I=1,NNN
      XBAR=XBAR+ARRAY(I,1)
      ZBAR=ZBAR+ARRAY(I,2)
      X1=X1+1.D0
124   CONTINUE
      XBAR=XBAR/X1
      ZBAR=ZBAR/X1
      RMEAN=DSORT(XBAR**2.D0+ZBAR**2.D0)
```



```
      VARZ=0.D0
      VARX=0.D0
      RVAR=0.D0
      DO 125,I=1,NNN
      VARZ=VARZ+(ARRAY(I,2)-ZBAR)**2.D0
      VARX=VARX+(ARRAY(I,1)-XBAR)**2.D0
      RVAR=RVAR+((DSORT(ARRAY(I,1)**2.D0+ARRAY(I,2)**2.D0)-
     1RMEAN)**2.D0)
125   CONTINUE
C IF ONLY ONE RAY GETS THROUGH, THEN X1 = 1 AND THE VARIANCE
C BLOWS UP.  THEREFORE, SET THE VARIANCE TO A VERY SMALL NUMBER
        IF ( X1 .EQ. 1.D0)THEN
        VARX = 1.D-15
        VARZ = 1.D-15
        RVAR = 1.D-15
         ELSE
         VARZ=VARZ/(X1-1.D0)
         VARX=VARX/(X1-1.D0)
         RVAR=RVAR/(X1-1.D0)
        ENDIF
      CALL INTENSE (VPSI1,VPP,DPM,DPMH,POWER,NNN,RMEAN,
     1RVAR,TA,POW,RMN,RV,PVR)
      WRITE (3,*)JTIME,RMEAN,RVAR,POWER,NNN
      RETURN
      END
C
C
      SUBROUTINE REFLECT(NCOMP,VP,VI,X,Z,II,XID,ROE,PMF,PME,VPSI,R)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VP(3),X(32),Z(32),XID(3,3),XMTX(3,3),XNO(3),ROE(3),
     1XN(3),VI(3),XNH(3),RM(3,3),R(3),VPSI(3)
C ESTABLISH THE FIRST COMPONENT (FOR PRINTING PURPOSES)
      NCOMP = 1.D0
C
C FIRST GET THE SEMI-LATUS RECTUM
      SLR = SLRECT(PMF,PME)
C CALCULATE MIRROR MATRIX
      CALL MIRROR(XID,PME,VPSI,XMTX)
C
      CALL NSUBO(SLR,VPSI,XNO)
C
C SINCE FINDING Z FOR ROE IS TRICKY, I SET SEPARATE SUBROUTINES
C FOR THE CALCULATION:
C FIRST CHECK TO SEE IF THE 2,2 COMPONENT OF THE MIRROR MATRIX
C IS ZERO; IF SO, THE PROBLEM OF FINDING THE Y-COMPONENT OF
C ROE REDUCES TO A FIRST ORDER PROBLEM
C
      IF(XMTX(2,2).NE.0.D0)THEN
         CALL ROEJOE2(VP,XMTX,XNO,X,Z,ROE,II)
         ELSE
         CALL ROEJOE1(VP,XMTX,XNO,X,Z,ROE,II)
         ENDIF
C
C FIND THE NORMAL VECTOR AT POINT ON SURFACE
C
      CALL NVECT(XNO,XMTX,ROE,XN)
C
C CALCULATE THE INCIDENT RAY VECTOR, THEN THE UNIT VECTOR
C THIS PROCEDURE IS REQUIRED ONLY FOR THE PRIMARY MIRROR;
C THEREFORE, IT IS NOT BROKEN OUT AS A SEPARATE SUBROUTINE
C
      DO 47 I = 1,3
 47   VI(I) = ROE(I) - VP(I)
      VIMAG = DSQRT(VI(1)**2.D0 + VI(2)**2.D0 + VI(3)**2.D0)
      DO 48 I = 1,3
 48   VI(I) = VI(I) / VIMAG
```

```
C
C FIND N-HAT (UNIT NORMAL) OF SURFACE, AT INCIDENT POINT
C
      CALL NHAT(XN,VI,XNH)
C CALCULATE THE REFLECTION MATRIX, RM
C
      CALL REFLMTX(XID,XNH,RM)
C
C AND , AT LAST, THE REFLECTED RAY DIRECTION IS:
C
      CALL RVECT(RM,VI,R)
C
      RETURN
      END
C
      SUBROUTINE COMPONENT2(NCOMP,VP,VI,II,E,F,VPSI,XID,ROE,R,SMM,
     1FLAG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPSI(3),XID(3,3),ROE(3),R(3),XNO(3),XMTX(3,3),
     1VI(3),VP(3),XN(3),XNH(3),RM(3,3)
      SLR = SLRECT(F,E)
      CALL MIRROR(XID,E,VPSI,XMTX)
      CALL NSUBO(SLR,VPSI,XNO)
C SOLVE FOR THE LENGTH OF THE INCOMING RAY VECTOR
      CALL LENGTH(II,NCOMP,VI,XMTX,VP,XNO,XLENGTH)
      CALL ROEVECT(XLENGTH,VP,VI,ROE)
C
C  ***  CHECK THE MAGNITUDE OF ROE VERSUS THE ALLOWABLE,
C  ***  DISCARD RAY IF IT DOESN'T STRIKE THE MIRROR
C
      XMAG=(ROE(1)**2.D0+ROE(2)**2.D0+ROE(3)**2.D0)**.5D0
      IF (XMAG.GT.SMM) THEN
      FLAG=-1.D0
      RETURN
      ELSE
      ENDIF
C
C   *** IF WITHIN LIMITS, CONTINUE.
      CALL NVECT(XNO,XMTX,ROE,XN)
      CALL NHAT(XN,VI,XNH)
      CALL REFLMTX(XID,XNH,RM)
      CALL RVECT(RM,VI,R)
C NOW R & ROE SHOW THE REFLECTED RAY
C RELATIVE TO THE SECOND COMPONENT VERTEX
      RETURN
      END
C
C
C  ******************************************************
C  ****  THIS ROUTINE CORRECTLY CALCULATES ALL THE  *****
C  ****  NECESSARY INFO TO GET THE POSITION AND     *****
C  ****  DIRECTION OF THE REFRACTED RAY             *****
C  ******************************************************
      SUBROUTINE COMPONENT3(NCOMP,VPL,VI,II,XLE1,XLF1,VPSI3,XID,
     1ROE,R,XNA,XNB,FLAG,XLM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPSI3(3),XID(3,3),ROE(3),R(3),XNO(3),XMTX(3,3),
     1VI(3),VPL(3),XN(3),XNH(3),RM(3,3)
      SLR=SLRECT(XLF1,XLE1)
      CALL MIRROR(XID,XLE1,VPSI3,XMTX)
      CALL NSUBO(SLR,VPSI3,XNO)
      CALL LENGTH(II,NCOMP,VI,XMTX,VPL,XNO,XLENGTH)
      CALL ROEVECT(XLENGTH,VPL,VI,ROE)
C
C  **** CHECK TO SEE IF RAY STRIKES SURFACE
      XMAG=(ROE(1)**2.+ROE(2)**2.+ROE(3)**2.)**.5
```



```
      IF(XMAG.GT.XLM)THEN
      FLAG=-1.DO
      RETURN
      ELSE
      ENDIF
      CALL NVECT(XNO,XMTX,ROE,XN)
      CALL NHAT(XN,VI,XNH)
      CALL REFRACT(VI,XNH,R,XNA,XNB)
      RETURN
      END
C
C  ******************************************************
C  *** THIS ROUTINE DETERMINES IF THE INCIDENT RAY FROM ***
C  *** THE SECONDARY MIRROR PASSES THROUGH THE HOLE IN  ***
C  *** THE PRIMARY MIRROR.  IF NOT, IT IS ELIMINATED    ***
C  *** FROM THE SYSTEM.                                 ***
C  ******************************************************
      SUBROUTINE COMPONENT4(NCOMP,VPL,VI,II,PME,PMF,VPSI1,
     1XID,PMHM,FLAG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPL(3),XID(3,3),VI(3),VPSI1(3),XMTX(3,3),
     1XNO(3),ROE(3)
      SLR=SLRECT(PMF,PME)
      CALL MIRROR(XID,PME,VPSI1,XMTX)
      CALL NSUBO(SLR,VPSI1,XNO)
      CALL LENGTH (II,NCOMP,VI,XMTX,VPL,XNO,XLENGTH)
      CALL ROEVECT(XLENGTH,VPL,VI,ROE)
      XMAG=(ROE(1)**2.DO+ROE(2)**2.DO+ROE(3)**2.DO)**.5DO
      IF (XMAG.GT.PMHM) THEN
      FLAG=-1.DO
      ELSE
      ENDIF
      RETURN
      END
C
C  ******************************************************
C  *****  THIS ROUTINE TAKES A GIVEN INCIDENT RAY     ****
C  *****  ,VI, AND A GIVEN SURFACE NORMAL AT POINT    ****
C  *****  OF INCIDENCE, XNH, WITH FIRST INDEX OF      ****
C  *****  REFRACTION, XNA, AND SECOND, XNB, AND RETURNS***
C  *****  A REFRACTED RAY DIRECTION, R. THIS ROUTINE  ****
C  *****  WILL CALCULATE THE REFRACTED RAY DIRECTION  ****
C  *****  FOR ANY SYSTEM AS LONG AS THE PROPER INFO   ****
C  *****  AS DEFINED ABOVE IS PASSED TO THE ROUTINE.  ****
C  *****  NOTE THAT THE DIRECTION OF THE NORMAL ON THE****
C  *****  SURFACE AT THE POINT OF INCIDENCE MUST HAVE ****
C  *****  THE PROPER DIRECTION WRT TO THE INCIDENT RAY****
C  *****  AS DEFINED IN THE REFRACTION EQUATION.      ****
C  ******************************************************
C
C  DEFINITIONS:
C  VI - A UNIT VECTOR ALONG THE DIRECTION OF THE INCIDENT RAY
C  XNH - A UNIT VECTOR ALONG THE SURFACE OF THE BOUNDARY
C        THIS VECTOR CAN TWO DIRECTIONS, THIS ROUTINE REQUIRES
C        XNH TO ALREADY BE DEFINED IN THE PROPER DIRECTION
C        WRT VI
C  R - A UNIT VECTOR TO REPRESENT THE DIRECTION OF THE
C      REFRATED RAY.
C  XNA - INDEX OF REFRACTION OF THE MEDIUM THE INCIDENT RAY
C        IS TRAVELLING IN
C  XNB - INDEX OF REFRACTION OF THE MEDIUM THE REFRACTED RAY
C        IS TO TRAVEL IN
C
      SUBROUTINE REFRACT(VI,XNH,R,XNA,XNB)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION R(3),VI(3),XNH(3),A(3)
```

```
C
C  ********  FIRST STEP IS TO CALCULATE SOME COEEFICIENTS
C  ******** BASED ON THE REFRACTION EQUATION
C
      XNC=XNA/XNB
      DO 1,I=1,3
1     A(I)=XNC*VI(I)
      B=1.-XNC**2.D0
      C=0.D0
      DO 2,I=1,3
2     C=C+XNH(I)*VI(I)
      E=XNC*C
      C=(C**2.D0)*(XNC**2.D0)
      D=((1.D0-(XNC**2.D0)+C)**.5D0)-E
C
C  *****  NOW THAT COEFFICIENTS ARE KNOWN, CALCULATE
C  *****  THE REFRACTED RAY DIRECTION
C
      DO 3, I=1,3
3     R(I)=A(I)-(B/D)*XNH(I)
      RETURN
      END
C
C
      SUBROUTINE LENGTH(II,NCOMP,VI,XMTX,VP,XNO,XLENGTH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VI(3),XMTX(3,3),VP(3),XNO(3),XJUNK(3)
C*******
C       OPEN(1,FILE='CHECK.DAT',STATUS='NEW')
C*******
      IF (NCOMP.EQ.3.D0) THEN
      XLENGTH=-VP(2)/VI(2)
      RETURN
      ELSE
      ENDIF
      DO 80 I=1,3
      XJUNK(I) = 0.
         DO 80 J=1,3
 80      XJUNK(I) = XJUNK(I) + XMTX(I,J) * VI(J)
      COEFFA = 0.D0
      DO 81 I=1,3
 81   COEFFA = COEFFA + VI(I) * XJUNK(I)
C
C SECOND COEFFICIENT
      DO 90 I=1,3
      XJUNK(I) = 0.D0
         DO 91 J=1,3
 91      XJUNK(I) = XJUNK(I) + XMTX(I,J) * VP(J)
 90   XJUNK(I) = XJUNK(I) + XNO(I)
      COEFFB = 0.D0
      DO 92 I=1,3
 92   COEFFB = COEFFB + 2.D0 * VI(I) * XJUNK(I)
C
C THIRD COEFFICIENT
      DO 95 I=1,3
      XJUNK(I) = 0.D0
         DO 96 J=1,3
 96      XJUNK(I) = XJUNK(I) + XMTX(I,J) * VP(J)
 95   XJUNK(I) = XJUNK(I) + 2.D0 * XNO(I)
      COEFFC = 0.D0
      DO 97 I=1,3
 97   COEFFC = COEFFC + VP(I) * XJUNK(I)
        IF (COEFFA.EQ.0.D0.AND.COEFFB.EQ.0.D0)THEN
        STOP
       ELSE
       ENDIF
```



```
      IF(COEFFA.EQ.0.D0)THEN
        XLENGTH = - COEFFC / COEFFB
        RETURN
      ELSE
        ENDIF
      IF(NCOMP.EQ.1.D0)THEN
      XLENGTH=(-COEFFB+DSQRT(COEFFB**2.D0-4.D0*COEFFA*COEFFC))/
     1(2.D0*COEFFA)
      ELSE
      XLENGTH=(-COEFFB-DSQRT(COEFFB**2.D0-4.D0*COEFFA*COEFFC))/
     1(2.D0*COEFFA)
      ENDIF
      RETURN
      END
C
      SUBROUTINE ROEVECT(XLENGTH,VP,VI,ROE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VP(3),VI(3),ROE(3)
      DO 110 I=1,3
 110  ROE(I) = VP(I) + XLENGTH * VI(I)
      RETURN
      END
C
      SUBROUTINE PARAMIN(VSAT,VPSI1,VPSI2,XL1,XL2,XL3,PMF,PME,
     1SMF,SME,XLF1,XLE1,XLF2,XLE2,DPM,DPMH,DSM,DL,VPSI3,VPSI4,XNA,XNB,
     2DIP,VPSI5,RES)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VSAT(3),VPSI1(3),VPSI2(3),XL1(3),
     1XL2(3),XL3(3),VPSI3(3),VPSI4(3),VPSI5(3)
      OPEN(1,FILE='STEVIE.DAT',STATUS='OLD')
      DO 10 I=1,3
C VP, VSAT ARE IN UNIVERSAL FRAME
 10   READ(1,*)VSAT(I),VPSI1(I),VPSI2(I)
      DO 11 J=1,3
 11   READ(1,*)XL1(J),XL2(J),XL3(J)
      READ(1,*)PMF,PME,SMF,SME,XLF1,XLE1
      READ(1,*)XLF2,XLE2,DPM,DPMH,DSM,DL
      DO 12 I=1,3
 12   READ(1,*)VPSI3(I),VPSI4(I)
      READ(1,*)XNA,XNB,DIP,RES
      READ(1,*)VPSI5(1),VPSI5(2),VPSI5(3)
      CLOSE(1)
      RETURN
      END
C
      SUBROUTINE PMTRANS(VPP,VSAT,VP)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPP(3),VSAT(3),VP(3)
C CONVERT VPP INTO THE PM FRAME
      DO 20 I=1,3
 20   VP(I) = VPP(I) - VSAT(I)
      RETURN
      END
C
      SUBROUTINE SMTRANS(XL,ROE,R,VI,VPS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XL(3),ROE(3),R(3),VI(3),VPS(3)
      DO 70 I=1,3
      VPS(I) = ROE(I) - XL(I)
 70   VI(I) = R(I)
      RETURN
      END
C
C
C  ************************************************************
C  *****  THIS SUBROUTINE TAKES A GIVEN REFLECTED RAY FROM ****
```

```
C   *****  THE SECONDARY MIRROR (R) AND A POSITION VECTOR OF****
C   *****  STRIKING POINT ON THE SM (ROE) AND CREATES TWO   ****
C   *****  NEW VECTORS, VI (THE NEW INCIDENT RAY FOR THE    ****
C   *****  LENS) AND VPS (VECTOR POINTING FROM VERTEX OF LENS***
C   *****  CONCAVE SIDE TO THE ORIGINATING POINT OF INCIDENT ***
C   *****  RAY).  XL IS A VECTOR LOCATING THE VERTEX OF THE ****
C   *****  SM WRT THE VERTEX OF THE LENS CONCAVE SIDE.      ****
C   ************************************************************
      SUBROUTINE LENSTRANS(XL,ROE,R,VI,VPS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XL(3),ROE(3),R(3),VI(3),VPS(3)
      DO 1 I=1,3
      VPS(I) = ROE(I) + XL(I)
 1    VI(I) = R(I)
      RETURN
      END
C
      SUBROUTINE XZCALC(DPM,DPMH,X,Z)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(32),Z(32)
C DIVIDE INTO 4 POINTS, EQUALLY SPACED, ALONG RADIUS
      RPM = DPM/2.D0
      RPMH = DPMH/2.D0
      RRAD = RPM - RPMH
      XINCR = RRAD/3.D0
      I = 1
      DO 30 RI = RPMH,RPM,XINCR
         DO 31 DEG = 90.D0,405.D0,45.D0
         X(I) = RI * DSIND(DEG)
         Z(I) = RI * DCOSD(DEG)
 31      I = I + 1
 30   CONTINUE
      RETURN
      END
C
      SUBROUTINE ROEJOE2(VP,XMTX,XNO,X,Z,ROE,II)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VP(3),XMTX(3,3),XNO(3),X(3),Z(3),ROE(3)
C
C CALCULATE THE QUADRATIC EQUATION COEFFICIENTS
C THESE COEFs WERE DERIVED FOR THE PM CASE ONLY
C
      B = ( X(II)*(XMTX(1,2)+XMTX(2,1))+Z(II)*(XMTX(2,3)
     1    + XMTX(3,2))+2.D0*XNO(2)) / XMTX(2,2)
      C = ( XMTX(1,1)*X(II)**2.D0+XMTX(3,3)*Z(II)**2.D0
     1    + X(II)*Z(II)*(XMTX(1,3)+XMTX(3,1))
     2    + 2.D0*(X(II)*XNO(1) + Z(II)*XNO(3)) ) / XMTX(2,2)
C
C THE QUADRATIC SOLUTION GIVES ROE(2), OR Y
C CHECK FOR A NEGATIVE UNDER THE RADICAL
C
      BSQM4C = B**2.D0 - 4.D0*C
      IF(BSQM4C.LT.0.D0)THEN
        PRINT*,'NEGATIVE UNDER RADICAL WHEN SOLVING FOR Y'
        STOP
      ELSE
        ENDIF
      ROE(2) = ( -B - DSQRT( BSQM4C ) ) / 2.D0
      ROE(1) = X(II)
      ROE(3) = Z(II)
C
      RETURN
      END
C
      SUBROUTINE ROEJOE1(VP,XMTX,XNO,X,Z,ROE,II)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```



```
      DIMENSION VP(3),XMTX(3,3),XNO(3),X(3),Z(3),ROE(3)
C
C CALCULATE THE FIRST-ORDER COEFFICIENTS FOR A*Y + B = 0.
C THESE COEFs WERE DERIVED FOR THE PM CASE ONLY
C
      A =  X(II)*(XMTX(1,2)+XMTX(2,1))+Z(II)*(XMTX(2,3)
     1    + XMTX(3,2))+2.D0*XNO(2)
      B =  XMTX(1,1)*X(II)**2.D0+XMTX(3,3)*Z(II)**2.D0
     1    + X(II)*Z(II)*(XMTX(1,3)+XMTX(3,1))
     2    + 2.D0*(X(II)*XNO(1) + Z(II)*XNO(3))
C
C
      ROE(2) = -B/A
      ROE(1) = X(II)
      ROE(3) = Z(II)
C
      RETURN
      END
C
      SUBROUTINE IDMATRX(XID)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XID(3,3)
C GENERATE AN IDENTITY MATRIX
      DO 40 I = 1,3
         DO 41 J=1,3
         XID(I,J) = 0.D0
         IF(I.EQ.J)XID(I,J) = 1.D0
 41      CONTINUE
 40   CONTINUE
      RETURN
      END
C
      FUNCTION SLRECT(F,E)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      SLRECT = F * (1.D0 + E )
      RETURN
      END
C
      SUBROUTINE MIRROR(XID,PME,VPSI,XMTX)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XID(3,3),VPSI(3),XMTX(3,3)
      DO 43 I = 1,3
         DO 43 J = 1,3
 43         XMTX(I,J) = XID(I,J) - (PME**2.D0) * VPSI(I) * VPSI(J)
      RETURN
      END
C
      SUBROUTINE NSUBO(SLR,VPSI,XNO)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPSI(3),XNO(3)
      DO 44 I = 1,3
 44   XNO(I) = -SLR * VPSI(I)
      RETURN
      END
C
      SUBROUTINE NVECT(XNO,XMTX,ROE,XN)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XNO(3),XMTX(3,3),ROE(3),XN(3)
      DO 45 I = 1,3
      SUM = 0.D0
         DO 46 K = 1,3
 46      SUM = SUM + XMTX(I,K) * ROE(K)
 45   XN(I) = XNO(I) + SUM
      RETURN
      END
C
```

```
      SUBROUTINE REFLMTX(XID,XNH,RM)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XID(3,3),XNH(3),RM(3,3)
      DO 55 I = 1,3
         DO 55 J = 1,3
 55        RM(I,J) = XID(I,J) - 2.D0*XNH(I)*XNH(J)
      RETURN
      END
C
      SUBROUTINE RVECT(RM,VI,R)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION RM(3,3),VI(3),R(3)
      DO 60 I = 1,3
      R(I) = 0.D0
         DO 61 J = 1,3
 61      R(I) = R(I) + RM(I,J)*VI(J)
 60   CONTINUE
      RETURN
      END
C
      SUBROUTINE NHAT(XN,VI,XNH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION XN(3),VI(3),XNH(3)
      XNMAG = DSQRT((XN(1)+1)**2.D0 + XN(2)**2.D0 + XN(3)**2.D0)
      DOT = 0.D0
      DO 50 I = 1,3
 50   DOT = DOT + VI(I)*XN(I)/XNMAG
      DO 51 I = 1,3
 51   XNH(I)   = -DSIGN(1.D0,DOT) * XN(I)/XNMAG
      RETURN
      END
C
C  **********  ***********************************************
C  ********  THIS ROUTINE WILL TAKE THE KNOWN        ********
C  ********  PARAMETERS OF AN OPTICAL ELEMENT,       ********
C  ********  NAMELY THE ECCENTRICITY,E, THE FOCAL    ********
C  ********  LENGTH,F, AND THE DIAMETER OF THE ELEMENT ******
C  ********  ,D, AND CALCULATES THE MAXIMUM MAGNITUDE********
C  ********  THE ROE VECTOR CAN HAVE IN ORDER TO     ********
C  ********  ACTUALLY STRIKE THE COMPONENT.          ********
C  ***********************************************************
      SUBROUTINE MAGNITUDE(F,E,D,XMAG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C  ****  IF E IS >1, THEN HYPERBOLOID
C
      IF(E.GT.1.D0) THEN
      A=F/E
      B=(F**2.D0-A**2.D0)**.5D0
      X=D/2.D0
      Y=((1+(X**2.D0)/(B**2.D0))*(A**2.D0))**.5D0-A
      XMAG=((X**2.D0)+(Y**2.D0))**.5D0
C
      ELSE
C  **** IF E = 1, THEN PARABOLA
      IF(E.EQ.1.D0)THEN
      X=D/2.D0
      Y=(X**2.D0)/(4.D0*F)
      XMAG=((X**2.D0)+(Y**2.D0))**.5D0
C  ****  OTHERWISE, ASSUMED TO BE SPHERICAL
C
      ELSE
      X=D/2.D0
      XMAG=((F-(F**2.D0-X**2.D0)**.5D0)**2.D0+X**2.D0)**.5D0
      ENDIF
      ENDIF
```



```
      RETURN
      END
C
      SUBROUTINE INTENSE (VPSI1,VPP,DPM,DPMH,POWER,NNN,RMEAN,
     1RVAR,TA,POW,RMN,RV,PVR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION VPSI1(3),VPP(3),VPU(3)
      PI=3.14159265358979300
C FIND MAGNITUDE OF VP
      VPMAG=DSQRT(VPP(1)**2.D0+VPP(2)**2.D0+VPP(3)**2.D0)
C GET UNIT VECTOR OF VPP
      DO 80 I=1,3
 80   VPU(I)=VPP(I)/VPMAG
C GET DOT PRODUCT OF VPP UNIT VECTOR AND VPSI1
        DOT = 0.D0
      DO 85 I=1,3
 85   DOT = DOT + VPU(I)*VPSI1(I)
C CALCULATE THE SUBTENDED PM AREA
      AP = DOT*PI*(DPM**2.D0-DPMH**2.D0)/4.D0
C THE TARGET LUMINANCE INTENSITY IS IN WATTS / STIR RADIAN:
      TARGI = 10.D0
C CALCULATE THE SOLID ANGLE
      OMEGA = AP/VPMAG**2.D0
C CALCULATE THE POWER COLLECTED BY THE PRIMARY MIRROR
      POWER = OMEGA * TARGI
C MULTIPLY POWER BY THE PERCENTAGE OF RAYS WHICH SUCESSFULLY TRACE
C TO THE DETECTOR
      PNNN = NNN
      PERC = PNNN/32.D0
      POWER = POWER * PERC
      POI = POWER/RVAR**2.D0
C
C RETAIN THE SUMMED VALUES OVER TIME FOR TIME=AVERAGE STATISTICS
C
      RMN = RMN + RMEAN
      RV = RV + RVAR
      TA = TA + RMEAN/RVAR
      POW = POW + POWER
      PVR = PVR + POI
      RETURN
      END
```

Appendix G. *OPTIMIZATION DATA*

This appendix contains the raw data from the optimization procedures.

Design Matrix

RUN	ONE	X1	X2	X3	X4	X1X1	X2X2	X3X3	X4X4
1	1	-1.000	-1.000	-1.000	-1.000	0.2	0.2	0.2	0.2
2	1	1.000	-1.000	-1.000	-1.000	0.2	0.2	0.2	0.2
3	1	-1.000	1.000	-1.000	-1.000	0.2	0.2	0.2	0.2
4	1	1.000	1.000	-1.000	-1.000	0.2	0.2	0.2	0.2
5	1	-1.000	-1.000	1.000	-1.000	0.2	0.2	0.2	0.2
6	1	1.000	-1.000	1.000	-1.000	0.2	0.2	0.2	0.2
	1	-1.000	1.000	1.000	-1.000	0.2	0.2	0.2	0.2
8	1	1.000	1.000	1.000	-1.000	0.2	0.2	0.2	0.2
9	1	-1.000	-1.000	-1.000	1.000	0.2	0.2	0.2	0.2
10	1	1.000	-1.000	-1.000	1.000	0.2	0.2	0.2	0.2
11	1	-1.000	1.000	-1.000	1.000	0.2	0.2	0.2	0.2
12	1	1.000	1.000	-1.000	1.000	0.2	0.2	0.2	0.2
13	1	-1.000	-1.000	1.000	1.000	0.2	0.2	0.2	0.2
14	1	1.000	-1.000	1.000	1.000	0.2	0.2	0.2	0.2
15	1	-1.000	1.000	1.000	1.000	0.2	0.2	0.2	0.2
16	1	1.000	1.000	1.000	1.000	0.2	0.2	0.2	0.2
17	1	0.000	0.000	0.000	0.000	-0.8	-0.8	-0.8	-0.8
18	1	-1.414	0.000	0.000	0.000	1.2	-0.8	-0.8	-0.8
19	1	1.414	0.000	0.000	0.000	1.2	-0.8	-0.8	-0.8
20	1	0.000	-1.414	0.000	0.000	-0.8	1.2	-0.8	-0.8
21	1	0.000	1.414	0.000	0.000	-0.8	1.2	-0.8	-0.8
22	1	0.000	0.000	-1.414	0.000	-0.8	-0.8	1.2	-0.8
23	1	0.000	0.000	1.414	0.000	-0.8	-0.8	1.2	-0.8
24	1	0.000	0.000	0.000	-1.414	-0.8	-0.8	-0.8	1.2
25	1	0.000	0.000	0.000	1.414	-0.8	-0.8	-0.8	1.2

Design Matrix

RUN	X1X2	X1X3	X1X4	X2X3	X2X4	X3X4	X1X2X3	X1X2X4	X1X3X4	X2X3X4	X1X2X3X4
1	1	1	1	1	1	1	-1	-1	-1	-1	1
2	-1	-1	-1	1	1	1	1	1	1	-1	-1
3	-1	1	1	-1	-1	1	1	1	-1	1	-1
4	1	-1	-1	-1	-1	1	-1	-1	1	1	1
5	1	-1	1	-1	1	-1	1	-1	1	1	-1
6	-1	1	-1	-1	1	-1	-1	1	-1	1	1
7	-1	-1	1	1	-1	-1	-1	1	1	-1	1
8	1	1	-1	1	-1	-1	1	-1	-1	-1	-1
9	1	1	-1	1	-1	-1	-1	1	1	1	-1
10	-1	-1	1	1	-1	-1	1	-1	-1	1	1
11	-1	1	-1	-1	1	-1	1	-1	1	-1	1
12	1	-1	1	-1	1	-1	-1	1	-1	-1	-1
13	1	-1	-1	-1	-1	1	1	1	-1	-1	1
14	-1	1	1	-1	-1	1	-1	-1	1	-1	-1
15	-1	-1	-1	1	1	1	-1	-1	-1	1	-1
16	1	1	1	1	1	1	1	1	1	1	1
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0

System Performance
Raw Data

Run	MTE	ILI	IVR	SM
1	0.00011231	.20503141819E+14	.11274439723E+08	269.40000
2	0.00005668	.14113037889E+15	.14258568398E+07	269.40000
3	0.00074052	.20621372305E+14	.15654312146E+08	269.40000
4	0.00032658	.14192188443E+15	.18638870395E+07	269.40000
5	0.00011188	.20503141819E+14	.11279491761E+08	717.00000
6	0.00005614	.14113037889E+15	.14264918659E+07	717.00000
7	0.00073853	.20621372302E+14	.15656680416E+08	717.00000
8	0.00032598	.14192188443E+15	.18639512800E+07	717.00000
9	0.00010127	.78574628053E+13	.57252572831E+08	756.40000
10	0.00006104	.52968952089E+14	.71723766015E+07	756.40000
11	0.00074262	.79054946725E+13	.79353365302E+08	756.40000
12	0.00032745	.53266019376E+14	.94536076915E+07	756.40000
13	0.00011161	.78574628063E+13	.57174717362E+08	1327.00000
14	0.00005628	.52968952086E+14	.71743014133E+07	1327.00000
15	0.00073888	.79041333645E+13	.79386995494E+08	1327.00000
16	0.00032608	.53266019373E+14	.94531345132E+07	1327.00000
17	0.00014496	.38423276402E+14	.87785084721E+07	650.10000
18	0.00070628	.57060202769E+13	.81051628739E+08	650.10000
19	0.00007760	.10438718353E+15	.28635238049E+07	650.10000
20	0.00003000	.38316462538E+14	.70199632657E+07	650.10000
21	0.00049245	.38639124144E+14	.96304683906E+07	650.10000
22	0.00014481	.38423276402E+14	.87845634729E+07	376.90000
23	0.00014521	.38423276402E+14	.87780725582E+07	1105.00000
24	0.00014488	.88453584216E+14	.30937068250E+07	399.50000
25	0.00014478	.21443293143E+14	.12052950502E+08	1194.00000

System Performance
Standardized Data

Run	MTE	ILI	IVR	SM
1	-0.64569	-0.68142	-0.34197	-1.36722
2	-0.86171	1.91325	-0.71279	-1.36722
3	1.79369	-0.67887	-0.17706	-1.36722
4	0.18632	1.93027	-0.69630	-1.36722
5	-0.64734	-0.68142	-0.34178	-0.07834
6	-0.86378	1.91325	-0.71277	-0.07834
7	1.78594	-0.67887	-0.17697	-0.07834
8	0.18400	1.93027	-0.69630	-0.07834
9	-0.68855	-0.95342	1.38921	0.03512
10	-0.84478	0.01692	-0.49642	0.03512
11	1.80182	-0.95239	2.22135	0.03512
12	0.18973	0.02331	-0.41053	0.03512
13	-0.64840	-0.95342	1.38627	1.67819
14	-0.86323	0.01692	-0.49635	1.67819
15	1.78732	-0.95242	2.22262	1.67819
16	0.18440	0.02331	-0.41055	1.67819
17	-0.51892	-0.29596	-0.43595	-0.27098
18	1.66072	-0.99970	2.28529	-0.27098
19	-0.78048	1.12291	-0.65866	-0.27098
20	-0.96529	-0.29826	-0.50216	-0.27098
21	0.83042	-0.29132	-0.40387	-0.27098
22	-0.51946	-0.29596	-0.43572	-1.05767
23	-0.51794	-0.29596	-0.43596	1.03893
24	-0.51920	0.78018	-0.64999	-0.99259
25	-0.51961	-0.66119	-0.31266	1.29521

MTE Empirical Model Statistics
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	4	22.94644	5.73661	114.344	0.0001
Error	21	1.05356	0.05017		
U Total	25	24.00000			

Root MSE	0.22399	R-square	0.9561
Dep Mean	-0.00000	Adj R-sq	0.9477
C.V.	-55996406434		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
X1	1	-0.534017	0.05008622	-10.662	0.0001
X2	1	0.825842	0.05008622	16.488	0.0001
X1X2	1	-0.351300	0.05599641	-6.274	0.0001
X1X1	1	0.451280	0.07919088	5.699	0.0001

ILI Empirical Model Statistics
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	5	23.92950	4.78590	1357.630	0.0001
Error	[illegible]	0.07050	0.00353		
U Total	2[illegible]	24.00000			

Root MSE	0.05937	R-square	0.9971
Dep Mean	0.00000	Adj R-sq	0.9963
C.V.	4947773259.1		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
X1	1	0.865107	0.01327667	65.160	0.0001
X4	1	-0.536822	0.01327667	-40.433	0.0001
X1X4	1	-0.407221	0.01484332	-27.435	0.0001
X1X1	1	0.185209	0.02099162	8.823	0.0001
X4X4	1	0.184153	0.02099162	8.773	0.0001

IVR Empirical Model Statistics
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	4	22.12887	5.53222	62.089	0.0001
Error	21	1.87113	0.08910		
U Total	25	24.00000			

Root MSE	0.29850	R-square	0.9220
Dep Mean	0.00000	Adj R-sq	0.9072
C.V.	74624641438		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
X1	1	-0.748866	0.06674832	-11.219	0.0001
X4	1	0.486955	0.06674832	7.295	0.0001
X1X4	1	-0.453307	0.07462464	-6.074	0.0001
X1X1	1	0.600368	0.10553518	5.689	0.0001

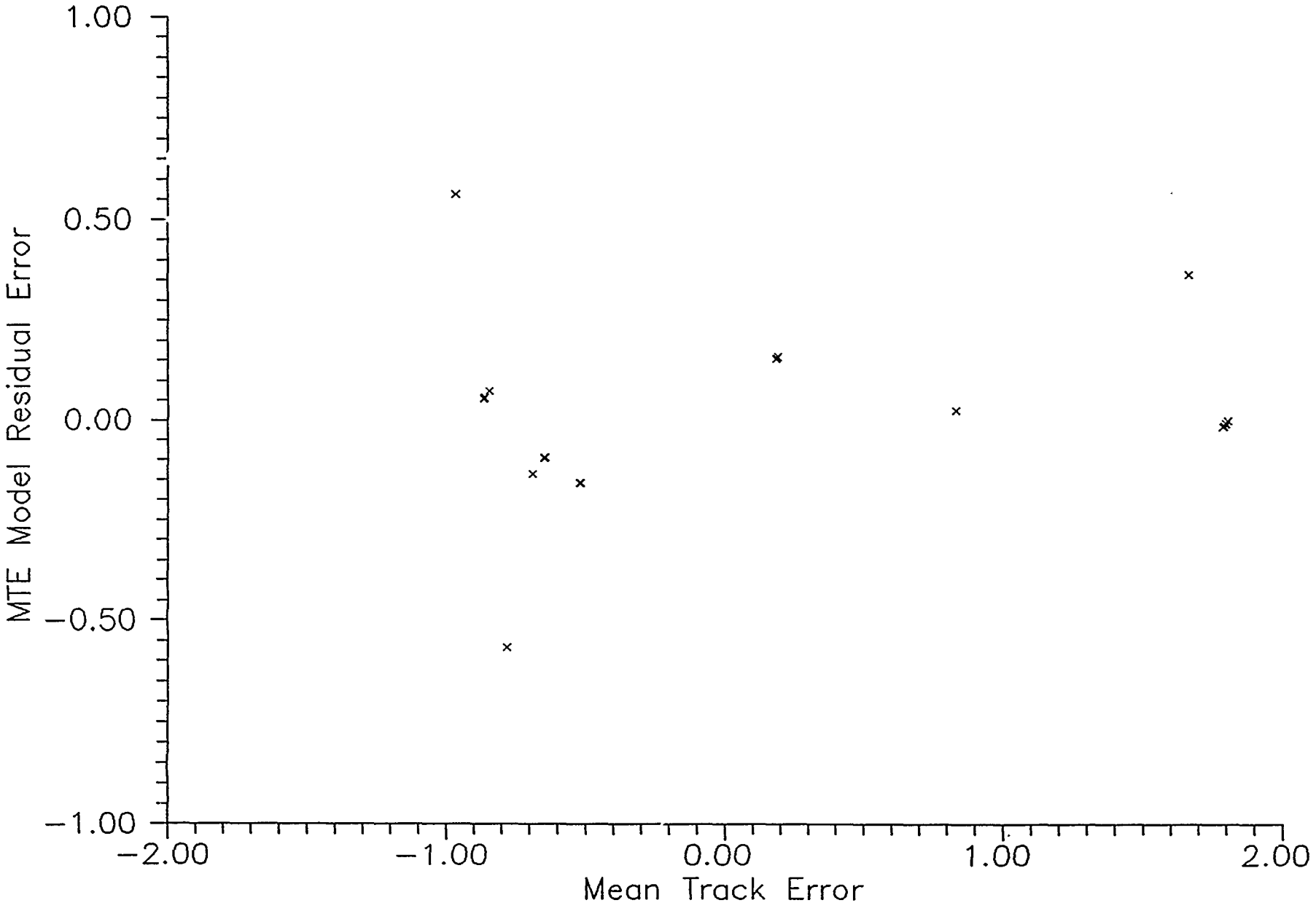
SM Empirical Model Statistics
Analysis of Variance

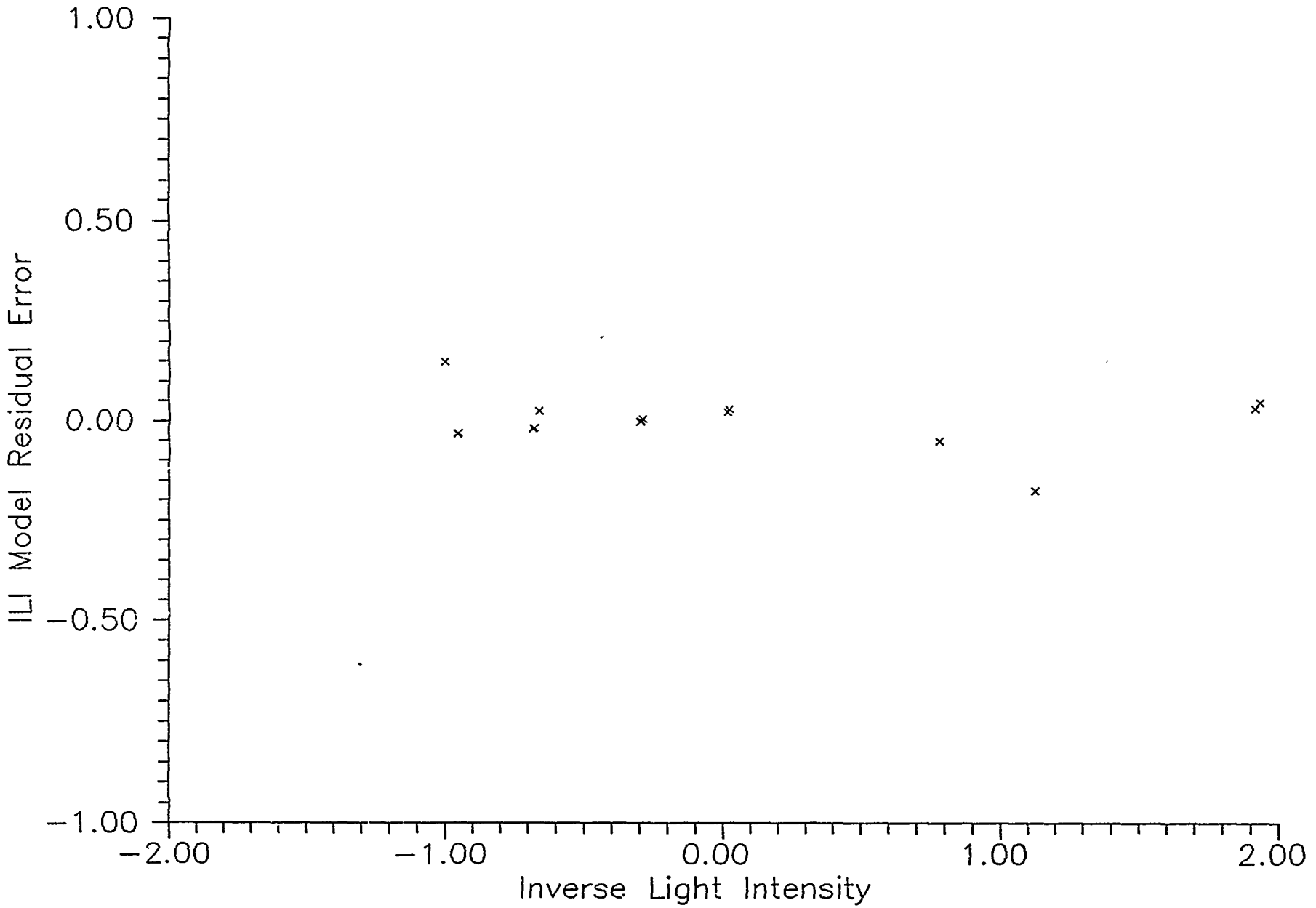
Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	5	23.99854	4.79971	65556.822	0.0001
Error	20	0.00146	0.00007		
U Total	25	24.00000			

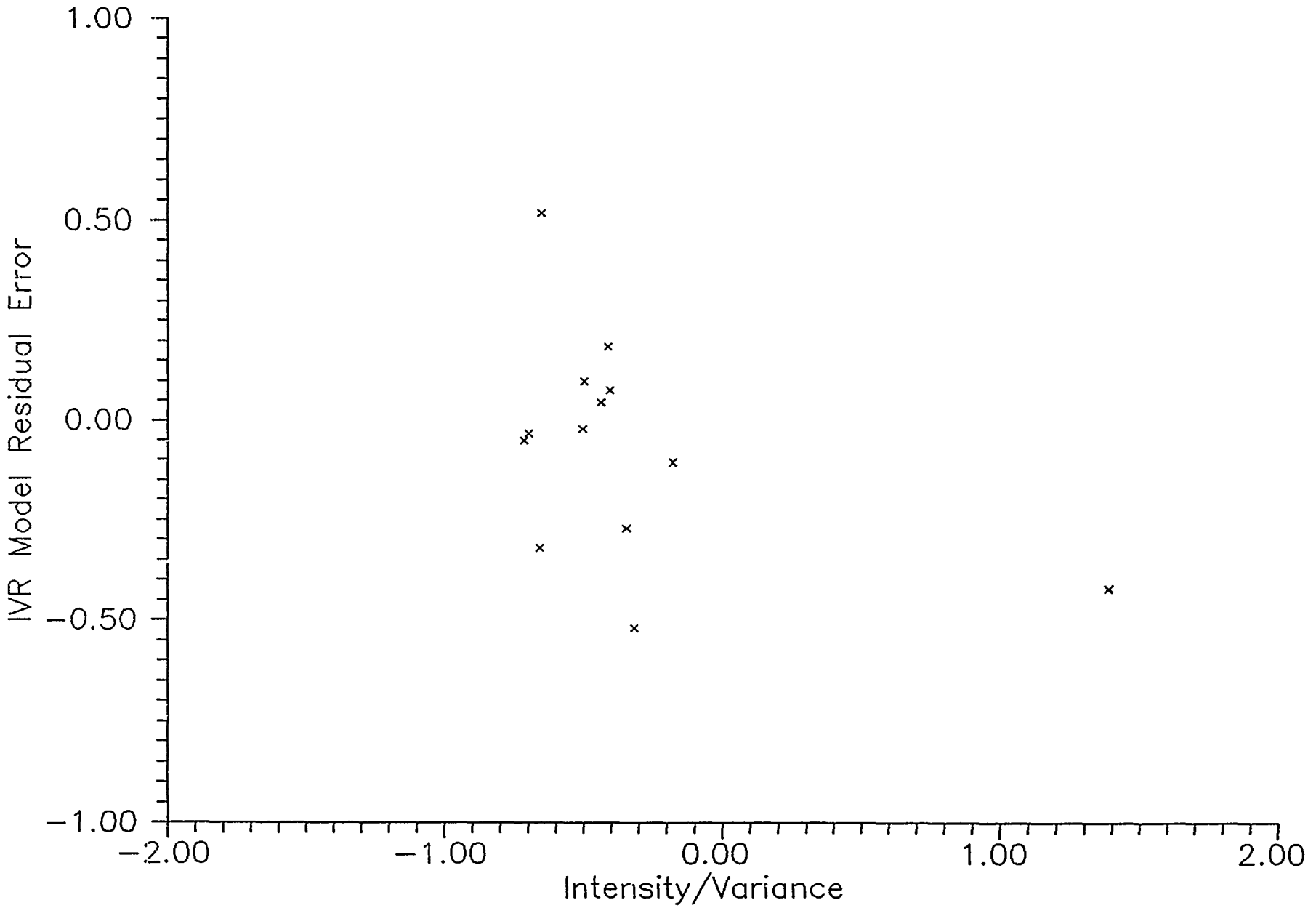
Root MSE	0.00856	R-square	0.9999
Dep Mean	0.00000	Adj R-sq	0.9999
C.V.	1.2845084E17		

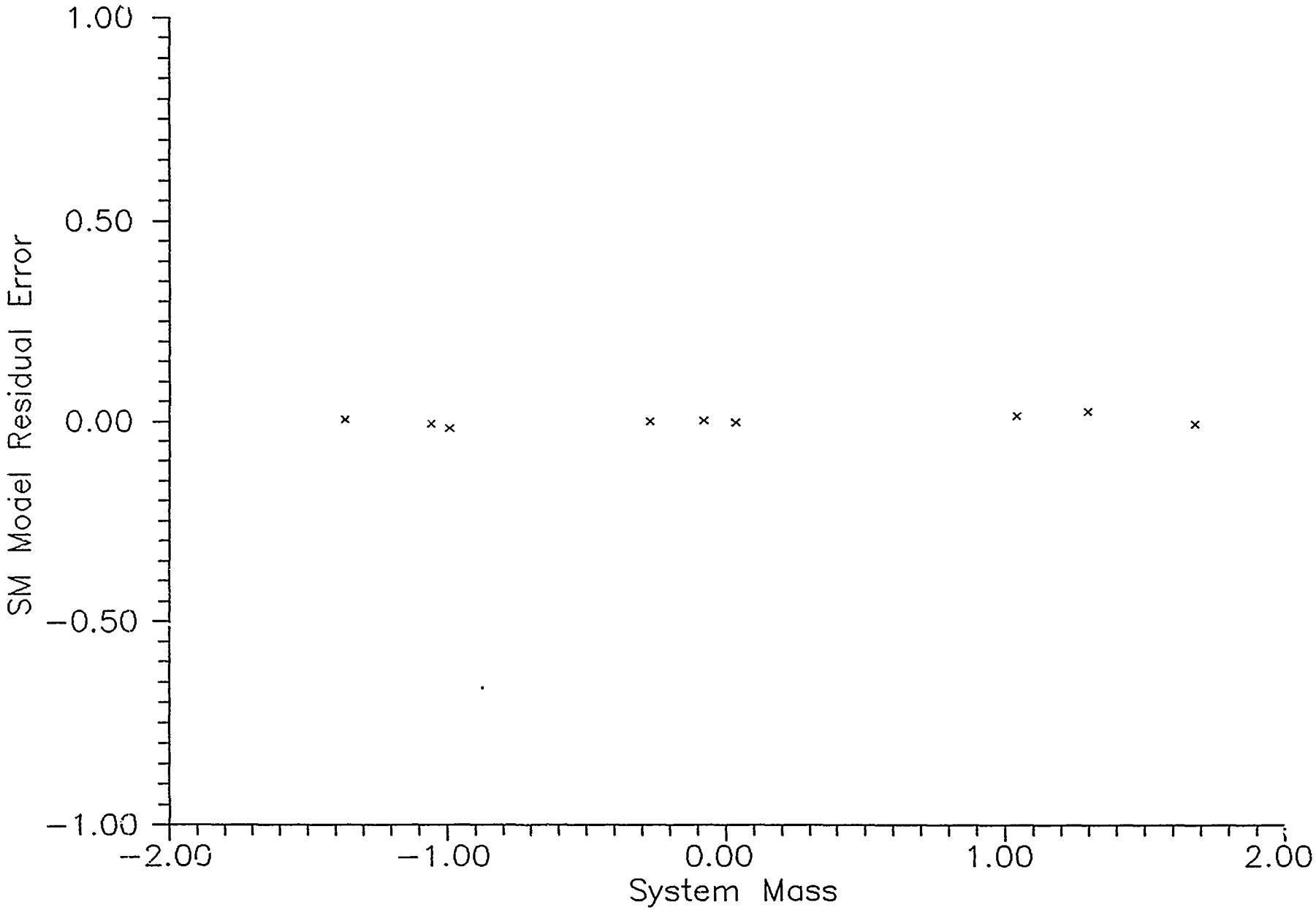
Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
X3	1	0.734666	0.00191336	383.967	0.0001
X4	1	0.793570	0.00191336	414.752	0.0001
X3X4	1	0.088546	0.00213914	41.393	0.0001
X3X3	1	0.129191	0.00302520	42.705	0.0001
X4X4	1	0.209531	0.00302520	69.262	0.0001


1.00
0.50
0.00
−0.50
−1.00
MTE Model Residual Error
−2.00
−1.00
0.00
1.00
2.00
Mean Track Error


1.00
0.50
0.00
−0.50
−1.00
ILI Model Residual Error
−2.00
−1.00
0.00
1.00
2.00
Inverse Light Intensity


1.00
0.50
0.00
−0.50
−1.00
IVR Model Residual Error
−2.00
−1.00
0.00
1.00
2.00
Intensity/Variance


1.00
0.50
0.00
−0.50
−1.00
SM Model Residual Error
−2.00
−1.00
0.00
1.00
2.00
System Mass

Selected design performance data. The following data is the model output with the factors determined in the optimization section. The angles below are the angles between the optical axis and the vector to the missile measured in radians.

Time	AZIMUTH ANGLE ERROR	ELEVATION ANGLE ERROR
0.000E+00	-3.5118E-01	4.7164E-01
1.000E+00	-5.1700E-03	-3.3489E-03
2.000E+00	-2.6005E-03	9.0887E-04
3.000E+00	4.9411E-04	-3.5709E-05
4.000E+00	2.2074E-04	-8.4027E-05
5.000E+00	-7.8905E-05	2.9800E-05
6.000E+00	-3.3159E-05	3.3892E-05
7.000E+00	-7.0588E-06	2.1898E-05
8.000E+00	-1.2233E-05	2.0189E-05
9.000E+00	-1.3584E-05	2.0178E-05
1.000E+01	-1.2277E-05	1.9266E-05
1.100E+01	-1.1536E-05	1.8235E-05
1.200E+01	-1.1044E-05	1.7338E-05
1.300E+01	-1.0514E-05	1.6501E-05
1.400E+01	-9.9941E-06	1.5697E-05
1.500E+01	-9.5064E-06	1.4931E-05
1.600E+01	-9.0436E-06	1.4203E-05
1.700E+01	-8.6026E-06	1.3511E-05
1.800E+01	-8.1830E-06	1.2852E-05
1.900E+01	-7.7840E-06	1.2225E-05
2.000E+01	-7.4044E-06	1.1629E-05
2.100E+01	-7.0434E-06	1.1062E-05
2.200E+01	-6.6999E-06	1.0523E-05
2.300E+01	-6.3732E-06	1.0010E-05
2.400E+01	-6.0625E-06	9.5218E-06
2.500E+01	-5.7668E-06	9.0576E-06
2.600E+01	-5.4856E-06	8.6160E-06
2.700E+01	-5.2181E-06	8.1960E-06
2.800E+01	-4.9637E-06	7.7964E-06
2.900E+01	-4.7216E-06	7.4164E-06
3.000E+01	-4.4914E-06	7.0549E-06
3.099E+01	-4.2724E-06	6.7110E-06
3.199E+01	-4.0640E-06	6.3839E-06
3.299E+01	-3.8659E-06	6.0727E-06
3.399E+01	-3.6773E-06	5.7767E-06
3.499E+01	-3.4980E-06	5.4952E-06
3.599E+01	-3.3274E-06	5.2274E-06
3.699E+01	-3.1652E-06	4.9727E-06
3.799E+01	-3.0108E-06	4.7303E-06
3.899E+01	-2.8640E-06	4.4998E-06
3.999E+01	-2.7243E-06	4.2806E-06
4.099E+01	-2.5914E-06	4.0720E-06
4.199E+01	-2.4650E-06	3.8736E-06
4.299E+01	-2.3448E-06	3.6849E-06
4.399E+01	-2.2304E-06	3.5054E-06
4.499E+01	-2.1216E-06	3.3347E-06
4.599E+01	-2.0181E-06	3.1723E-06
4.699E+01	-1.9197E-06	3.0178E-06
4.799E+01	-1.8260E-06	2.8708E-06
4.898E+01	-1.7369E-06	2.7311E-06
4.998E+01	-1.6522E-06	2.5981E-06
5.098E+01	-1.5715E-06	2.4716E-06
5.198E+01	-1.4949E-06	2.3513E-06
5.298E+01	-1.4219E-06	2.2369E-06
5.398E+01	-1.3525E-06	2.1280E-06
5.498E+01	-1.2865E-06	2.0245E-06
5.598E+01	-1.2237E-06	1.9260E-06

5.698E+01	-1.1639E-06	1.8323E-06
5.798E+01	-1.1071E-06	1.7432E-06
5.898E+01	-1.0531E-06	1.6585E-06
5.998E+01	-1.0016E-06	1.5778E-06
6.098E+01	-9.5270E-07	1.5012E-06
6.198E+01	-9.0616E-07	1.4282E-06
6.298E+01	-8.6189E-07	1.3588E-06
6.398E+01	-8.1977E-07	1.2928E-06
6.498E+01	-7.7972E-07	1.2301E-06
6.598E+01	-7.4161E-07	1.1704E-06
6.697E+01	-7.0535E-07	1.1136E-06
6.797E+01	-6.7086E-07	1.0596E-06
6.897E+01	-6.3804E-07	1.0082E-06
6.997E+01	-6.0683E-07	9.5930E-07
7.097E+01	-5.7713E-07	9.1281E-07
7.197E+01	-5.4888E-07	8.6860E-07
7.297E+01	-5.2201E-07	8.2654E-07
7.397E+01	-4.9644E-07	7.8654E-07
7.497E+01	-4.7212E-07	7.4849E-07
7.597E+01	-4.4898E-07	7.1230E-07
7.697E+01	-4.2696E-07	6.7788E-07
7.797E+01	-4.0602E-07	6.4514E-07
7.897E+01	-3.8610E-07	6.1401E-07
7.997E+01	-3.6715E-07	5.8439E-07
8.097E+01	-3.4912E-07	5.5622E-07
8.197E+01	-3.3196E-07	5.2944E-07
8.297E+01	-3.1565E-07	5.0396E-07
8.397E+01	-3.0012E-07	4.7973E-07
8.496E+01	-2.8535E-07	4.5668E-07
8.596E+01	-2.7130E-07	4.3476E-07
8.696E+01	-2.5793E-07	4.1392E-07
8.796E+01	-2.4521E-07	3.9410E-07
8.896E+01	-2.3311E-07	3.7524E-07
8.996E+01	-2.2160E-07	3.5732E-07
9.096E+01	-2.1065E-07	3.4027E-07
9.196E+01	-2.0023E-07	3.2406E-07
9.296E+01	-1.9032E-07	3.0864E-07
9.396E+01	-1.8089E-07	2.9398E-07
9.496E+01	-1.7191E-07	2.8004E-07
9.596E+01	-1.6337E-07	2.6679E-07
9.696E+01	-1.5525E-07	2.5419E-07
9.796E+01	-1.4752E-07	2.4221E-07
9.896E+01	-1.4017E-07	2.3082E-07
9.996E+01	-1.3318E-07	2.1999E-07
1.010E+02	-1.2652E-07	2.0969E-07
1.020E+02	-1.2019E-07	1.9991E-07
1.030E+02	-1.1416E-07	1.9061E-07
1.040E+02	-1.0843E-07	1.8176E-07
1.050E+02	-1.0297E-07	1.7336E-07
1.060E+02	-9.7781E-08	1.6537E-07
1.070E+02	-9.2841E-08	1.5779E-07
1.080E+02	-8.8141E-08	1.5057E-07
1.090E+02	-8.3669E-08	1.4372E-07
1.100E+02	-7.9413E-08	1.3721E-07
1.110E+02	-7.5363E-08	1.3103E-07
1.120E+02	-7.1510E-08	1.2516E-07
1.129E+02	-6.7842E-08	1.1958E-07
1.139E+02	-6.4352E-08	1.1429E-07
1.149E+02	-6.1030E-08	1.0926E-07
1.159E+02	-5.7869E-08	1.0448E-07
1.169E+02	-5.4860E-08	9.9955E-08
1.179E+02	-5.1996E-08	9.5658E-08
1.189E+02	-4.9271E-08	9.1581E-08
1.199E+02	-4.6676E-08	8.7714E-08
1.209E+02	-4.4206E-08	8.4048E-08
1.219E+02	-4.1855E-08	8.0573E-08

1.229E+02	-3.9616E-08	7.7280E-08
1.239E+02	-3.7485E-08	7.4161E-08
1.249E+02	-3.5456E-08	7.1207E-08
1.259E+02	-3.3524E-08	6.8412E-08
1.269E+02	-3.1683E-08	6.5769E-08
1.279E+02	-2.9931E-08	6.3269E-08
1.289E+02	-2.8086E-08	6.0650E-08
1.300E+02	-2.6725E-08	5.8759E-08
1.310E+02	-2.5195E-08	5.6655E-08
1.320E+02	-2.3746E-08	5.4672E-08
1.330E+02	-2.2369E-08	5.2805E-08
1.340E+02	-2.1055E-08	5.1048E-08
1.350E+02	-1.9803E-08	4.9398E-08
1.360E+02	-1.8608E-08	4.7849E-08
1.370E+02	-1.7469E-08	4.6398E-08
1.380E+02	-1.6382E-08	4.5041E-08
1.390E+02	-1.5345E-08	4.3774E-08
1.400E+02	-1.4354E-08	4.2594E-08
1.410E+02	-1.3409E-08	4.1498E-08
1.420E+02	-1.2505E-08	4.0483E-08
1.430E+02	-1.1642E-08	3.9546E-08
1.441E+02	-1.0817E-08	3.8685E-08
1.451E+02	-1.0027E-08	3.7897E-08
1.461E+02	-9.2714E-09	3.7180E-08
1.471E+02	-8.5476E-09	3.6533E-08
1.481E+02	-7.8539E-09	3.5954E-08
1.491E+02	-7.1887E-09	3.5441E-08
1.501E+02	-6.5502E-09	3.4992E-08
1.511E+02	-5.9367E-09	3.4608E-08
1.521E+02	-8.2159E-09	-2.4732E-08
1.531E+02	-8.6360E-09	-2.2716E-08
1.541E+02	-7.9604E-09	-2.1417E-08
1.551E+02	-7.4815E-09	-2.0270E-08
1.561E+02	-7.0281E-09	-1.9173E-08
1.571E+02	-6.5946E-09	-1.8128E-08
1.582E+02	-6.1815E-09	-1.7133E-08
1.592E+02	-5.7878E-09	-1.6187E-08
1.602E+02	-5.4125E-09	-1.5286E-08
1.612E+02	-5.0546E-09	-1.4428E-08
1.622E+02	-4.7134E-09	-1.3611E-08
1.632E+02	-4.3879E-09	-1.2834E-08
1.642E+02	-4.0775E-09	-1.2093E-08
1.652E+02	-3.7813E-09	-1.1388E-08
1.662E+02	-3.4987E-09	-1.0717E-08
1.672E+02	-3.2291E-09	-1.0077E-08
1.682E+02	-2.9716E-09	-9.4683E-09
1.692E+02	-2.7259E-09	-8.8882E-09
1.702E+02	-2.4912E-09	-8.3357E-09
1.712E+02	-2.2671E-09	-7.8093E-09
1.723E+02	-2.0530E-09	-7.3077E-09
1.733E+02	-1.8485E-09	-6.8298E-09
1.743E+02	-1.6530E-09	-6.3744E-09
1.753E+02	-1.4661E-09	-5.9404E-09
1.763E+02	-1.2873E-09	-5.5266E-09
1.773E+02	-1.1164E-09	-5.1322E-09
1.783E+02	-9.5281E-10	-4.7562E-09
1.793E+02	-7.9626E-10	-4.3976E-09
1.803E+02	-6.4638E-10	-4.0555E-09
1.813E+02	-5.0283E-10	-3.7293E-09
1.823E+02	-3.6529E-10	-3.4180E-09
1.833E+02	-2.3347E-10	-3.1210E-09
1.843E+02	-1.0708E-10	-2.8374E-09
1.853E+02	1.4150E-11	-2.5668E-09
1.864E+02	1.3049E-10	-2.3083E-09
1.874E+02	2.4219E-10	-2.0615E-09
1.884E+02	3.4947E-10	-1.8256E-09

1.894E+02	4.5257E-10	-1.6003E-09
1.904E+02	5.5169E-10	-1.3848E-09
1.914E+02	6.4705E-10	-1.1789E-09
1.924E+02	7.3883E-10	-9.8182E-10
1.934E+02	8.2722E-10	-7.9329E-10
1.944E+02	9.1240E-10	-6.1283E-10
1.954E+02	9.9453E-10	-4.4003E-10
1.964E+02	1.0738E-09	-2.7450E-10
1.974E+02	1.1503E-09	-1.1586E-10
1.984E+02	1.2242E-09	3.6246E-11
1.994E+02	1.2957E-09	1.8215E-10
2.005E+02	1.3648E-09	3.2218E-10
2.015E+02	1.4318E-09	4.5664E-10
2.025E+02	1.4966E-09	5.8583E-10
2.035E+02	1.5595E-09	7.1002E-10
2.045E+02	1.6206E-09	8.2949E-10
2.055E+02	1.6799E-09	9.4448E-10
2.065E+02	1.7376E-09	1.0552E-09
2.075E+02	1.7937E-09	1.1620E-09
2.085E+02	1.8483E-09	1.2650E-09
2.095E+02	1.9016E-09	1.3644E-09
2.105E+02	1.9536E-09	1.4604E-09
2.115E+02	2.0043E-09	1.5533E-09
2.125E+02	2.0539E-09	1.6431E-09
2.135E+02	2.1025E-09	1.7302E-09
2.145E+02	2.1500E-09	1.8145E-09
2.156E+02	2.1967E-09	1.8964E-09
2.166E+02	2.2424E-09	1.9760E-09
2.176E+02	2.2873E-09	2.0533E-09
2.186E+02	2.3315E-09	2.1286E-09
2.196E+02	2.3750E-09	2.2019E-09
2.206E+02	2.4179E-09	2.2735E-09
2.216E+02	2.4601E-09	2.3433E-09
2.226E+02	2.5018E-09	2.4116E-09
2.236E+02	2.5431E-09	2.4785E-09
2.246E+02	2.5838E-09	2.5440E-09
2.256E+02	2.6242E-09	2.6083E-09
2.266E+02	2.6642E-09	2.6714E-09
2.276E+02	2.7039E-09	2.7335E-09
2.286E+02	2.7434E-09	2.7946E-09
2.297E+02	2.7825E-09	2.8549E-09
2.307E+02	2.8215E-09	2.9144E-09
2.317E+02	2.8603E-09	2.9731E-09
2.327E+02	2.8990E-09	3.0313E-09
2.337E+02	2.9376E-09	3.0889E-09
2.347E+02	2.9762E-09	3.1461E-09
2.357E+02	3.0147E-09	3.2028E-09
2.367E+02	3.0532E-09	3.2592E-09
2.377E+02	3.0917E-09	3.3154E-09
2.387E+02	3.1303E-09	3.3714E-09
2.397E+02	3.1690E-09	3.4272E-09
2.407E+02	3.2079E-09	3.4829E-09
2.417E+02	3.2469E-09	3.5387E-09
2.427E+02	3.2860E-09	3.5945E-09
2.438E+02	3.3254E-09	3.6504E-09
2.448E+02	3.3650E-09	3.7064E-09
2.458E+02	3.4049E-09	3.7627E-09
2.468E+02	3.4451E-09	3.8192E-09
2.478E+02	3.4856E-09	3.8761E-09
2.488E+02	3.5265E-09	3.9333E-09
2.498E+02	3.5677E-09	3.9910E-09
2.508E+02	3.6094E-09	4.0492E-09
2.518E+02	3.6515E-09	4.1078E-09
2.528E+02	3.6940E-09	4.1671E-09
2.538E+02	3.7370E-09	4.2270E-09
2.548E+02	3.7806E-09	4.2876E-09

2.558E+02	3.8247E-09	4.3489E-09
2.568E+02	3.9154E-09	4.3142E-09
2.579E+02	3.9631E-09	4.3612E-09
2.589E+02	4.0104E-09	4.4250E-09
2.599E+02	4.0568E-09	4.4898E-09
2.609E+02	4.1038E-09	4.5555E-09
2.619E+02	4.1517E-09	4.6223E-09
2.629E+02	4.2004E-09	4.6902E-09
2.639E+02	4.2499E-09	4.7592E-09
2.649E+02	4.3002E-09	4.8294E-09
2.659E+02	4.3514E-09	4.9009E-09
2.669E+02	4.4034E-09	4.9737E-09
2.679E+02	4.4564E-09	5.0479E-09
2.689E+02	4.5103E-09	5.1235E-09
2.699E+02	4.5652E-09	5.2006E-09
2.709E+02	4.6212E-09	5.2793E-09
2.720E+02	4.6783E-09	5.3595E-09
2.730E+02	4.7364E-09	5.4414E-09
2.740E+02	4.7958E-09	5.5251E-09
2.750E+02	4.8563E-09	5.6106E-09
2.760E+02	4.9181E-09	5.6980E-09
2.770E+02	4.9812E-09	5.7873E-09
2.780E+02	5.0456E-09	5.8787E-09
2.790E+02	5.1114E-09	5.9721E-09
2.800E+02	5.1787E-09	6.0677E-09
2.810E+02	5.2474E-09	6.1656E-09
2.820E+02	5.3178E-09	6.2659E-09
2.830E+02	5.3897E-09	6.3686E-09
2.840E+02	5.4633E-09	6.4738E-09
2.850E+02	5.5387E-09	6.5816E-09
2.861E+02	5.6158E-09	6.6922E-09
2.871E+02	5.6949E-09	6.8055E-09
2.881E+02	5.7758E-09	6.9219E-09
2.891E+02	5.8588E-09	7.0412E-09
2.901E+02	5.9439E-09	7.1637E-09
2.911E+02	6.0311E-09	7.2895E-09
2.921E+02	6.1206E-09	7.4186E-09
2.931E+02	6.2125E-09	7.5513E-09
2.941E+02	6.3068E-09	7.6877E-09
2.951E+02	6.4036E-09	7.8279E-09
2.961E+02	6.5030E-09	7.9720E-09
2.971E+02	6.6052E-09	8.1202E-09
2.981E+02	6.7102E-09	8.2727E-09
2.991E+02	6.8182E-09	8.4297E-09
3.002E+02	6.9293E-09	8.5913E-09
3.012E+02	7.0435E-09	8.7577E-09
3.022E+02	7.1611E-09	8.9291E-09
3.032E+02	7.2822E-09	9.1058E-09
3.042E+02	7.4070E-09	9.2879E-09
3.052E+02	7.5355E-09	9.4756E-09
3.062E+02	7.6679E-09	9.6693E-09
3.072E+02	7.8045E-09	9.8691E-09
3.082E+02	7.9454E-09	1.0075E-08
3.092E+02	8.0908E-09	1.0288E-08
3.102E+02	8.2408E-09	1.0508E-08
3.112E+02	8.3958E-09	1.0736E-08
3.122E+02	8.5559E-09	1.0971E-08
3.132E+02	8.7214E-09	1.1214E-08
3.143E+02	8.8924E-09	1.1465E-08
3.153E+02	9.0694E-09	1.1725E-08
3.163E+02	9.2525E-09	1.1995E-08
3.173E+02	9.4421E-09	1.2274E-08
3.183E+02	9.6385E-09	1.2563E-08
3.193E+02	9.8420E-09	1.2862E-08
3.203E+02	1.0053E-08	1.3173E-08
3.213E+02	1.0272E-08	1.3496E-08

3.223E+02	1.0499E-08	1.3831E-08
3.233E+02	1.0735E-08	1.4179E-08
3.243E+02	1.0980E-08	1.4540E-08
3.253E+02	1.1234E-08	1.4916E-08
3.263E+02	1.1499E-08	1.5307E-08
3.273E+02	1.1775E-08	1.5714E-08
3.283E+02	1.2062E-08	1.6138E-08
3.294E+02	1.2360E-08	1.6579E-08
3.304E+02	1.2672E-08	1.7040E-08
3.314E+02	1.2997E-08	1.7520E-08
3.324E+02	1.3336E-08	1.8022E-08
3.334E+02	1.3691E-08	1.8546E-08
3.344E+02	1.4061E-08	1.9094E-08
3.354E+02	1.4448E-08	1.9667E-08
3.364E+02	1.4854E-08	2.0267E-08
3.374E+02	1.5278E-08	2.0895E-08
3.384E+02	1.5723E-08	2.1554E-08
3.394E+02	1.6191E-08	2.2246E-08
3.404E+02	1.6681E-08	2.2972E-08
3.414E+02	1.7197E-08	2.3736E-08
3.424E+02	1.7739E-08	2.4539E-08
3.435E+02	1.8310E-08	2.5385E-08
3.445E+02	1.8912E-08	2.6277E-08
3.455E+02	1.9547E-08	2.7218E-08
3.465E+02	2.0218E-08	2.8212E-08
3.475E+02	2.0927E-08	2.9263E-08
3.485E+02	2.1677E-08	3.0376E-08
3.495E+02	2.2473E-08	3.1555E-08
3.505E+02	2.3317E-08	3.2807E-08
3.515E+02	2.4214E-08	3.4137E-08
3.525E+02	2.5169E-08	3.5552E-08
3.535E+02	2.6186E-08	3.7061E-08
3.545E+02	2.7271E-08	3.8671E-08
3.555E+02	2.8431E-08	4.0392E-08
3.565E+02	2.9673E-08	4.2234E-08
3.576E+02	3.1006E-08	4.4211E-08
3.586E+02	3.2437E-08	4.6335E-08
3.596E+02	3.3978E-08	4.8623E-08
3.606E+02	3.3020E-08	4.3806E-08
3.616E+02	-6.4080E-08	-1.1887E-07

Appendix H. *RESPONSE SURFACE MODELS FOR PERFORMANCE MEASURES*

$$MTE = -0.534017f_1 + 0.825842f_2 - 0.35130f_1f_2 + 0.45128(f_1^2 - 0.8)$$

Optimum MTE = −1.5295

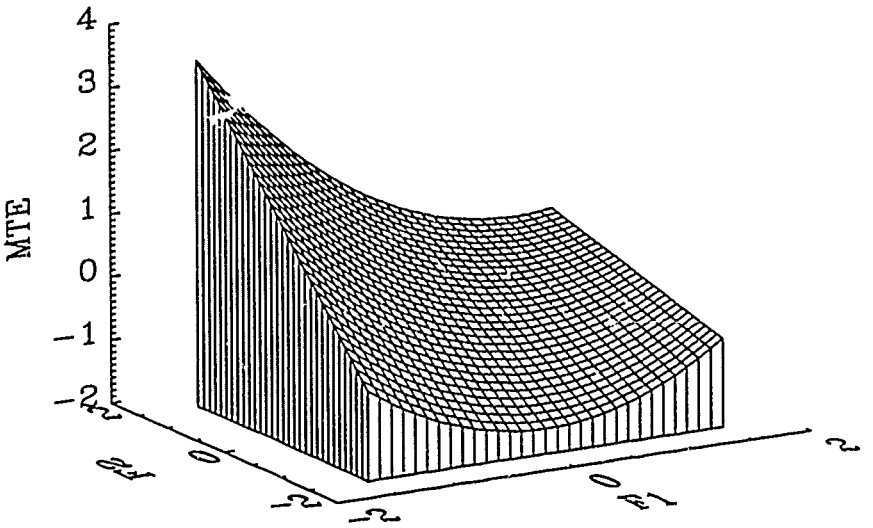


Figure H.1. Mean Track Error

$$ILI = 0.865107f_1 - 0.536822f_4 - 0.407221f_1f_4 + 0.185209(f_1^2 - 0.8)$$

Optimum ILI = −1.1505

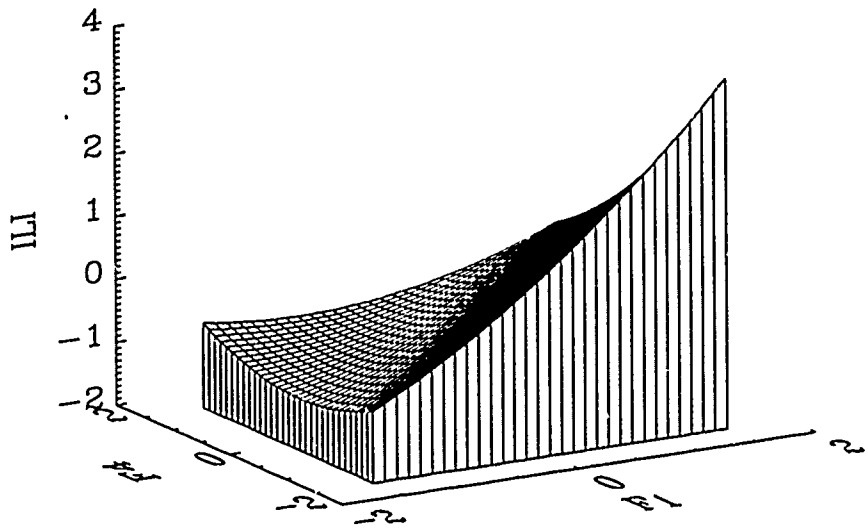


Figure H.2. Inverse Light Intensity

$$IVR = -0.748866 f_1 + 0.486955 f_4 - 0.453307 f_1 f_4 + 0.600368 (f_1^2 - 0.8)$$

Optimum IVR = 3.3739

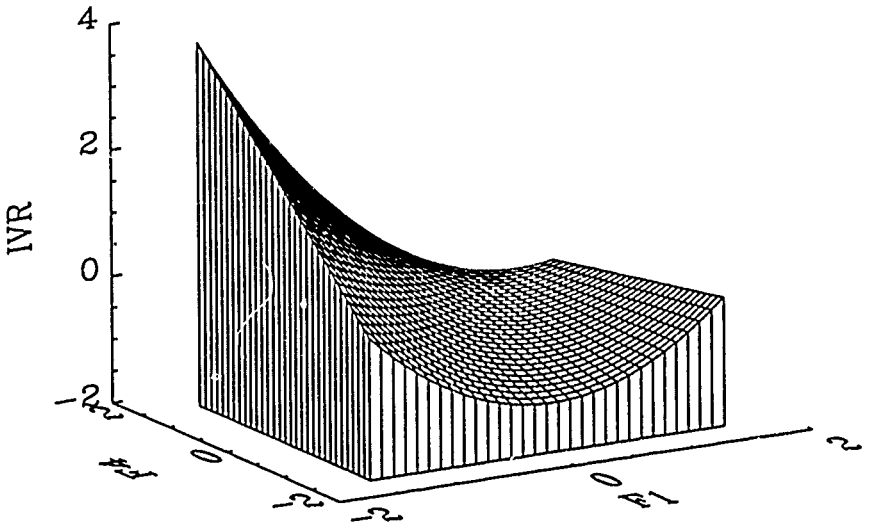


Figure H.3. Intensity Over Variance

$$SM = 0.734666 f_3 + 0.793570 f_4 + 0.08546 f_3 f_4 + 0.129191 (f_3^2 - 0.8) + 0.209531 (f_4^2 - 0.8)$$

Optimum SM = −1.5776

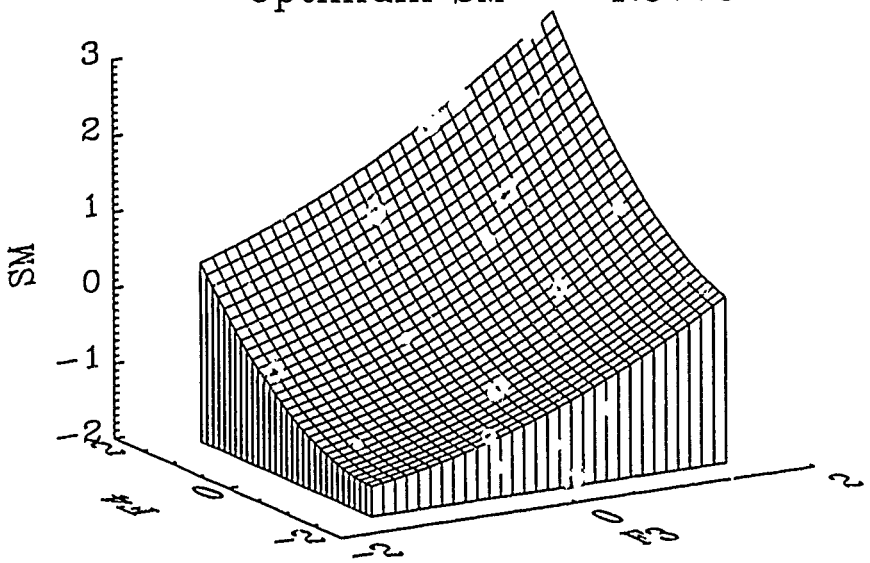


Figure H.4. System Mass

Appendix I. *MODEL FLEXIBILITY*

This paper has already shown the ability of bond graphs to interface different systems. Bond graphs also lend themselves very well to changes in the system configuration. Because bond graphs utilize energy relationships between systems, as the system changes or expands, it is simply a matter of determining where the new element inserts in the system and defining the energy relationship between it and the existing system. To demonstrate this, four areas of model expansion are addressed in this chapter: flexible optics, thermal effects, passive vibration control, and active vibration control.

I.1 FLEXIBLE OPTICS

The design and development of flexible optical space platforms will undoubtedly require a vibration sensitivity analysis. The system model contains the ability to determine the effects of vibrations on the optics as a system. However, the model does not allow displacements between the different optical components. Currently, the only motion permitted is the motion of the entire optical subsystem with respect to the system structure. A more accurate model of the system may be achieved by allowing motion between the different optical components. The modeling of this relative motion is referred to as the "flexible optics" portion of the dynamics.

The system model adapts easily to this change. The vibrations model contains all the dynamical equations of the structure. If the optical structure is allowed to become flexible (instead of rigid), the structural vibrations model must be changed. The original structures model was based on the NASTRAN analysis of the system. A new system would be modeled, adding a flexible subsystem of beams to connect the optical components (see Figure I.1). The optical components – primary mirror, secondary mirror, lens and detector plate - are still considered rigid bodies. Therefore, they would be modeled as separate lump masses, connected to the optic substructure

by rigid elements and supported by a flexible system of beams. The output transformation block in the structural vibrations bond graph model is basically a matrix multiplication to retrieve the position of the structural node of interest. This output transformation gives the position and orientation of that node in global coordinates.

In order to obtain information about the optical component attachment nodes, the output transformation block must be modified to include data on these attachment nodes. With the required information, it is possible to locate the vertex of each optical component. The position and orientation of each optical component vertex are required as inputs to the optics program for ray tracing.

The optics program utilizes a vector that describes the orientation of the optical component. This vector, ψ, is a unit vector pointing from the vertex towards the focus. This unit vector is defined in a base frame. In terms of the system, this base frame is the body frame. If the ψ vectors can be determined for each component, then the orientation of the components with respect to each other can also be determined.

In tracing the ray from one component to the next, the optics program uses the ψ vectors to determine the orientations of the components and also uses position vectors to locate the vertices of the components. Position information is available as an output from the structures model on each of the components. Thus the position vectors of all the components can be determined through simple vector addition.

By providing all the ψ vector information as well as position information on the components, the optics program can easily incorporate vibrations in terms of displacements in the ray trace calculations. The required changes to the system model are minimal. The only change to the system model is incorporation of the structural information for a new NASTRAN analysis into the vibrations bond graph model and expansion of the output transformation to include information on the additional optical components' attachment points.

Traditional ray trace software packages can perform the same basic function

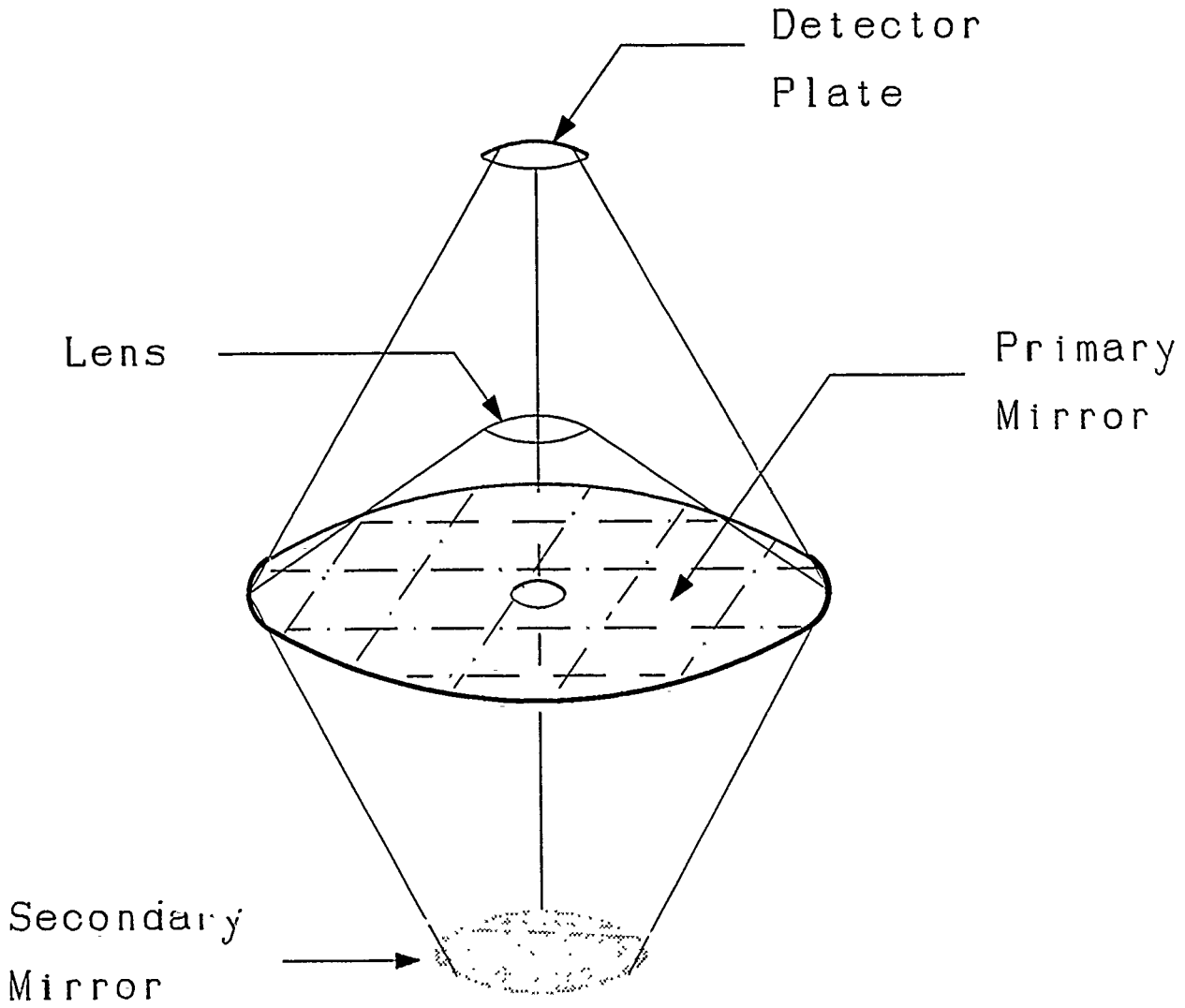


Figure I.1. Optic structure subsystem.

However, these packages lack the ability to do the same function in a simulation environment. These packages can provide a means of aligning optical components off axis, but are normally done manually. Sensitivity analyses using these software packages do not include dynamic interactions with the rest of the system. The advantage of using the dynamic relations in the flexible optics is that a direct correlation can be made between the optical performance and the vibrations present in the structure. Thus performance measures on the allowable vibrations in the system can be defined easier and earlier in the design.

I.2 THERMAL EFFECTS

It is possible for the effects of thermal radiation to be a factor in the design of space based optical systems. The effect of structural heating due to thermal radiation manifest itself as a displacement in unconstrained beams. Temperature displacements in the beam may be due to either uniform changes in temperature or to differential changes. A uniform change refers to a temperature change that is constant throughout the beam. A uniform change causes the beam to increase or decrease in length. All other dimensions of the beam will change proportionately, but only the change in length is significant in most cases. A differential change means that the top and bottom of the beam are subjected to different temperatures, while the average temperature remains unchanged. The member will not change in length but will undergo a curvature of its longitudinal axis.

A uniform temperature change results in a change in the length of the beam by the amount:

$$\Delta L = \alpha L(\Delta T) \tag{I.1}$$

in which ΔL is the change in length, α is the coefficient of thermal expansion, L is the length of the beam, and ΔT is the uniform change in temperature

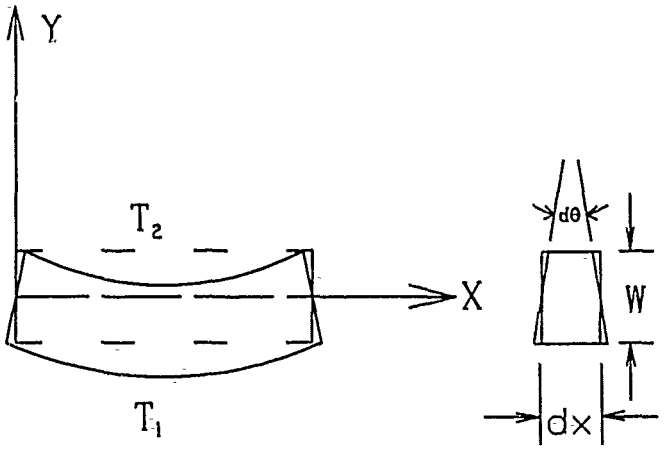


Figure I.2. Temperature deformation

(Gere, J. and Weaver, W. 1965).

For a differential temperature change, where the temperature varies linearly between T_2 at the top of the beam and a higher temperature T_1 at the bottom, the relative angle of rotation between the sides of the element as shown in Figure I.2 is:

$$d\theta = \frac{\alpha(T_1 - T_2)dx}{W} \tag{I.2}$$

in which W is the depth of the beam. Using Equation I.1 and I.2, the beam displacements can be calculated (Gere, J. and Weaver, W. 1965).

In modeling these thermal effects, the thermal displacements in a beam produces two phenomena. First, there is a relative velocity between any two points on the beam during the expansion or curving process. This velocity is a function of the rate of expansion or curving. In most cases, the rate of expansion or curving is so small that this induced velocity is negligible. The second phenomenon is the effect the displacement has on the moment arm characteristics of the beam. For example, look at the simple system of Figure I.3. The system consists of a beam of mass M and moment of inertia J supported at each end by a spring and damper assembly. Consider small oscillations of the beam in the vertical directions and no movement

in the X direction so that:

$$V_A \approx V - a\omega \tag{I.3}$$

$$V_B \approx V + b\omega \tag{I.4}$$

The system bond graph is shown in Figure I.4. The moment arm characteristics of the beam is modeled using transformers TF_1 and TF_2. The modulus of transformer TF_1 is a and the modulus of transformer TF_2 is b. To incorporate the displacement effects of structural heating, both transformers become modulated transformers where the modulus now becomes a function of the thermal environment. In the case of uniform temperature changes, the modulus of TF_1 becomes

$$a + \Delta a = a(1 + \alpha\Delta T) \tag{I.5}$$

and the modulus of TF_2 becomes

$$b + \Delta b = b(1 + \alpha\Delta T) \tag{I.6}$$

Similar results are obtained in the case of differential changes in temperature. Thus, thermal displacements affect only the transformers. All other portion of the bond graph remain the same.

With an understanding of the thermal environment of the system, both the uniform temperature change (ΔT) and the differential change in temperature ($T_1 - T_2$) can be defined for each beam as functions of time and/or other system parameters. The effects of structural heating due to thermal radiation can then be incorporated into the model with the use of modulated transformers.

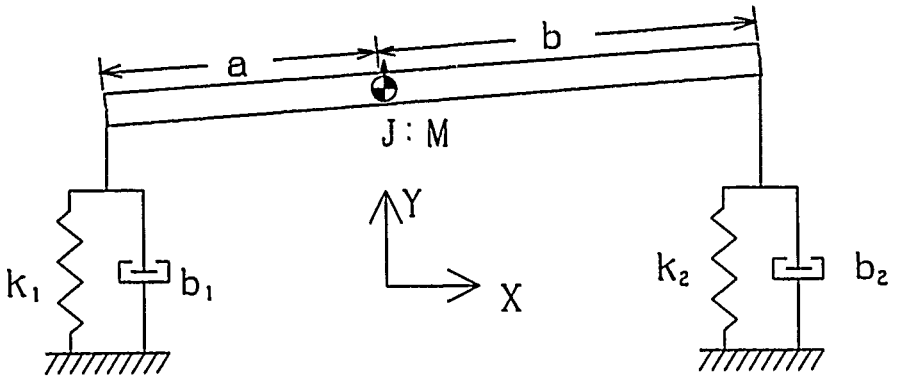


Figure I.3. Example beam

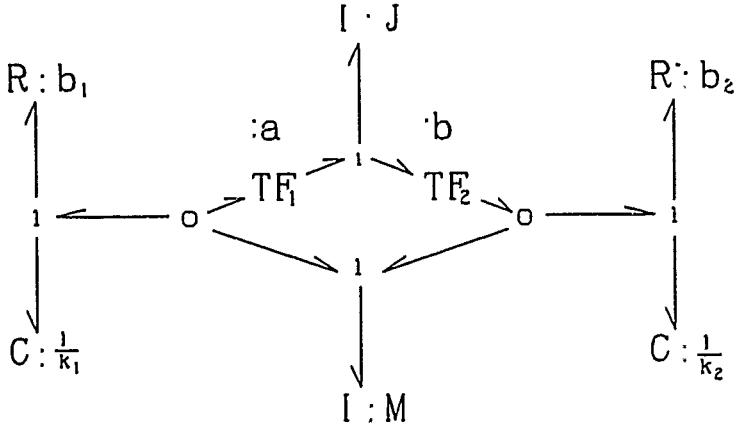


Figure I.4. Example beam bond graph

I.3 PASSIVE VIBRATION CONTROL

In the current system model, no attempt was made to isolate the optical assembly from vibration in the outer structure. The system, including the optics, was modeled as one body. The orbital dynamics model and the rigid body rotational model combine to model the rigid body characteristics while the structural vibrations model captures its flexible characteristics. In the NASTRAN analysis, the connections between the optical assembly and the outer structure are modeled as welded joints. However, the system model can be expanded to include some form of passive vibration control at the connection points. The most effective way to add passive vibration control is to model the outer support structure and the optical assembly as two bodies connected by spring and damper assemblies. This would require modifications to both the structural vibrations model and the rigid body rotational dynamics model. The orbital dynamics model would remain as is in order to model the orbital characteristics of the system center of mass.

The modification to the structural vibrations model would require a new NASTRAN analysis that excludes the optical assembly. The NASTRAN analysis would provide modal information on only the outer structure and allow for inputs at the controller locations and the optical assembly connections. The current vibration model bond graph would be expanded to accept the three additional inputs from the optical assembly. The model would use the new modal parameters provided from the NASTRAN analysis and, thus, model vibration in the outer structure only.

The rigid body rotational dynamics bond graph would be expanded to model the full rigid body dynamics of both the outer structure and the optical assembly. It would also model the spring and damper assemblies that connect the two bodies. The bond graph would model all six degrees of freedom (DOF) of each body (three linear velocities and three angular velocities for each). Using only linear springs, the size and complexity of the rigid body model would increase from the current three state model (three inertial masses) to a 21 state model (six inertial masses,

six linear masses and nine linear springs). If torsional springs are also modeled, an additional nine states will be added. To demonstrate both the concept and the increase in complexity, Figure I.5 shows a planar one DOF example of the proposed expansion. The system in the example consists of an inner mass M_1 connected to an outer structure M_2 by two spring and damper assemblies. Motion is allowed in the X direction only. Figure I.6 shows the corresponding bond graph for this one DOF system. If the same structure is given three DOF (rotation about Z and linear motion in X and Y), the bond graph expands to that shown in Figure I.7. The full six DOF rigid body dynamics model bond graph would expand to more than twice the size of Figure I.7 and would include gyrator coupling between the inertial elements (Pacejka, H. 1985).

The example discussed treats the outer structure as a flexible body but treats the optical assembly as rigid. If it becomes necessary to model the optical assembly as a flexible body, a second vibration model can be developed for the optical assembly. This would require a NASTRAN analysis of the optical assembly and a duplication of the current vibration model bond graph.

In summarizing the above discussion, the original one body model was expanded into a two body model connected by spring and damper assemblies. In doing so, the bond graphs that modeled the dynamics of the one body system were duplicated and added to the original model to form a two body system. The same procedure can be repeated until every beam, momentum wheel, mirror, and component is modeled individually. This allows for expansion of the model to whatever complexity is necessary to answer the questions of interest.

I.4 ACTIVE VIBRATION CONTROL

One large item of interest in the control of large flexible space structures is active control of structural vibrations. Once again the problem of not having a fixed frame of reference from which to apply forces is present. Thus any attempt in actively

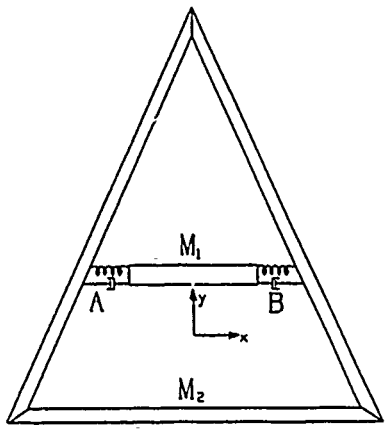


Figure I.5. Planar 1 DOF example

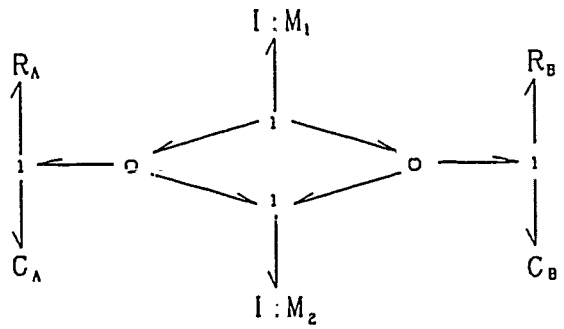


Figure I.6. Example 1 DOF bond graph

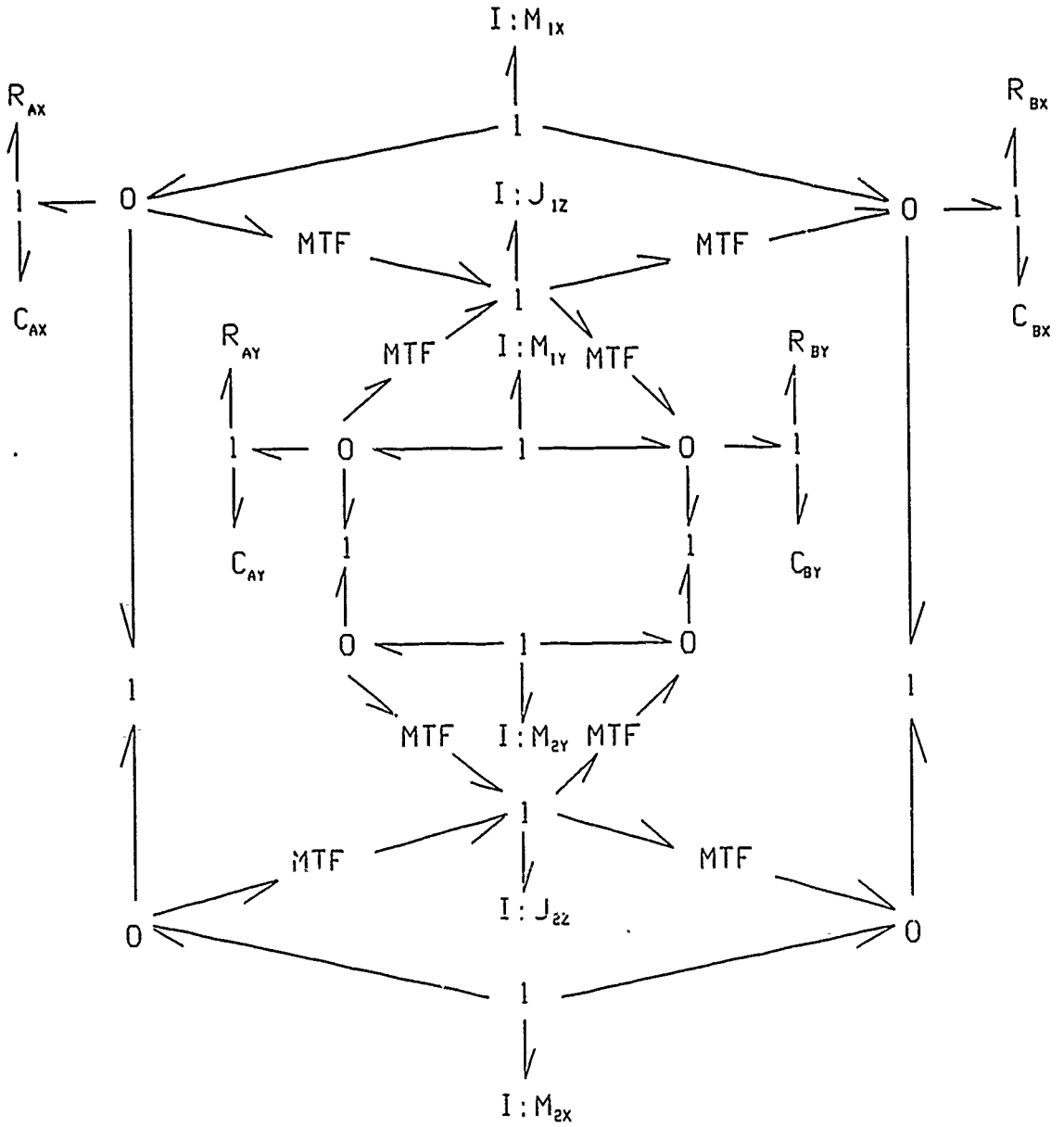


Figure I.7. Example 3 DOF bond graph

controlling vibrations will result in the actuator being mounted on the structure itself. The difficulty is that when the structure moves, the actuator moves with it, resulting in dynamic coupling between the actuator and the structure. In order to design an effective control algorithm, this coupling must be accounted for. Many research projects are involved in modeling this coupling and determining optimal placements of the actuators on the structure as well as sensor placement.

The techniques utilized in this study can be easily extended to active vibration control of the type mentioned. As an illustration, a proof-mass actuator system will be modeled using bond graphs and will be integrated into the system model developed earlier.

I.4.1 Proof-Mass Controller Many of the actuators proposed for vibration control in space use the principle of momentum exchange. Since the actuator is mounted on the structure, all of the forces are internal to the actuator/structural system. This means that the momentum of the system has to be conserved; a change in momentum in one part of the system has to be matched by an opposing change in momentum in another part for a net change of zero. Specifically, a change in the momentum of the proof mass must be balanced by a change in the momentum of the structure. The development here will concentrate on a linear proof mass actuator.

The physical shape of a typical linear proof mass actuator is shown in Figure I.8. The device is built around a linear electromagnetic motor. The coil is rigidly mounted on a bracket that fixes the actuator to the structure. Also connected to the bracket is a rigid frame which supports the springs, which in turn are coupled to the framework. The springs provide a returning force to the proof mass and transfer force to the beam.

The proof mass consists of a rectangular ring along with a central steel member. The central member passes through the coil and restricts movement of the proof mass to a single axis. Magnets are attached to the inside of the ring such that the

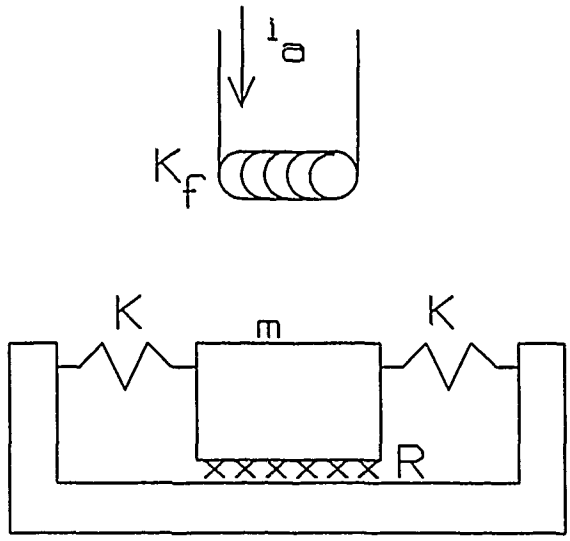


Figure I.8. Linear proof mass actuator.

interaction of the magnets with the current in the coil results in a force on the proof mass. As with the momentum wheels, a force input on the proof mass reacts with the structure in the opposite direction. If the force input to the proof mass can be made proportional to the structural vibrations, the vibrations in the structure can be controlled by transferring the vibrations to the proof mass isolating the structural deformations.

The first step in applying the techniques of active vibration control with proof mass actuators in the satellite system is the development of the associated bond graph. It is sometimes simpler to realize the bond graph of a physical system if the free body diagram of the system is first drawn. This is the case for the proof mass actuators.

The free body diagram (FBD) is shown in Figure I.9. As seen in the FBD, the force seen by the structure is equal to the applied force minus the force in the springs minus the loss due to friction. This simplifies the problem of determining the bond graph. With known equations, the bond graph can now be constructed.

The bond graph of the proof mass controller is shown in Figure I.10. This bond graph also incorporates the ability of bond graphs to easily incorporate dynamics of

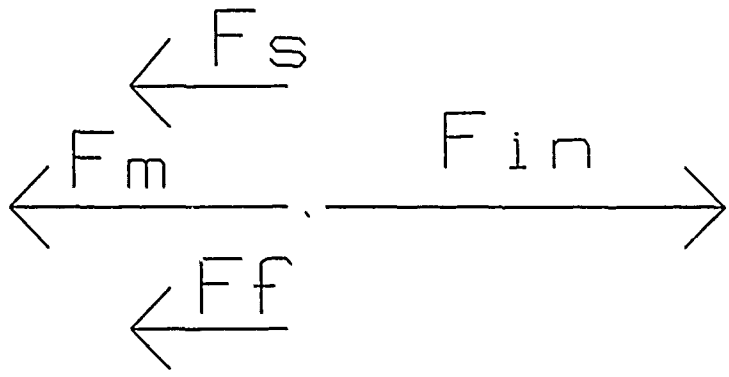


Figure I.9. Proof mass free body diagram.

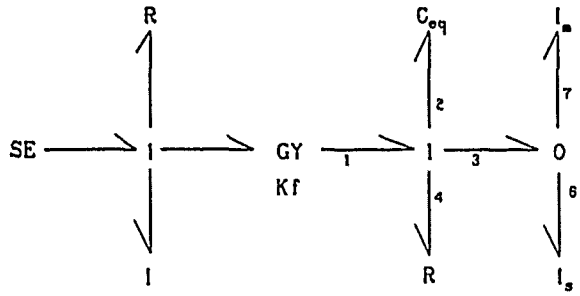


Figure I.10. Proof mass controller bond graph.

different types of systems, in this case electrical and mechanical. The applied force to the proof mass is supplied by an electromagnetic motor. The electrical current in the primary windings is converted to a mechanical force on the proof mass through the magnets in the ring. Thus a conversion (or transformation) is made from an electrical flow signal to a mechanical effort signal. The conversion from a flow signal to an effort signal is accomplished with a gyrator as shown in Figure I.10. The modulus for the gyrator is equal to the force constant of the motor.

The two springs on the proof mass can be collapsed into one equivalent spring. This is done only to simplify the bond graph and does not lose any generalities. The springs can be combined in a manner similar to combining parallel resistors in an electrical network:

$$C_{eq} = \frac{C_1 C_2}{C_1 + C_2} \tag{I.7}$$

The viscous damping, R, and the spring equivalent, C_{eq}, are placed on a 1 junction as they share a common velocity. The resulting effort signal is then joined at a 0 node with the mass components of the proof mass and the associated mass component of the beam. Summing the effort signals at this point reveals that the bond graph is equivalent to the FBD:

$$\dot{P}_7 = E_1 - \frac{q_2}{C_{eq}} - R(\frac{P_6}{I_6} - \frac{P_7}{I_7}) \tag{I.8}$$

where,

E_1 = motor torque (output of gyrator)

$\dot{P}_7$ = effort on beam (derivative of momentum)

q_2 = displacement of springs from rest position

P_6 = momentum of proof mass

R = coefficient of friction

I_6 and I_7 = mass of proof mass and beam, respectively.

The state equations for this system are:

$$\begin{bmatrix} \dot{q}_2 \\ \dot{P}_6 \\ \dot{P}_7 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{I_6} & \frac{-1}{I_7} \\ \frac{-1}{C_{eq}} & \frac{-R}{I_6} & \frac{R}{I_7} \\ \frac{-1}{C_{eq}} & \frac{-R}{I_6} & \frac{R}{I_7} \end{bmatrix} \begin{bmatrix} q_2 \\ P_6 \\ P_7 \end{bmatrix} + E1 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{I.9}$$

Integrating this bond graph with the system is shown in Figure I.11. The active controls model interfaces with the vibrations model. The modal acceleration of the structure can be easily determined from the forces applied to the modal masses. The modal accelerations are then transformed to generalized accelerations which are then input to the proof mass controller. In return, the resulting active control forces upon the structure must be sent through transformers to get modal forces. The

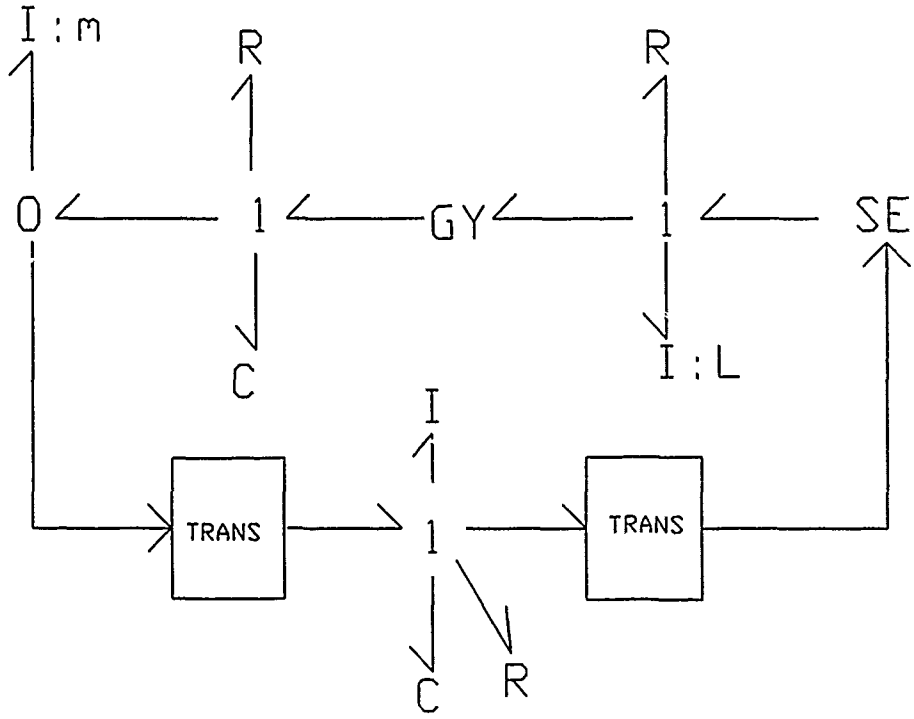


Figure I.11. Controller/structure interface.

resultant modal forces, due to active vibration control, are summed with the modal forces due to momentum wheel control. The summed modal forces are then applied to the vibrations model to determine the motion of the flexible structure.

These same formulations can be accomplished using traditional engineering techniques. However, the technique presented here offers the design engineer the ability to determine approximate optimal locations for sensors and proof mass controllers early in the design process. With this knowledge and the ability to analyze the system with all dynamical interactions present, the designer possesses the ability to determine approximately how effective the active controllers will be in damping out structural vibrations. As such, performance measures on other parts of the system can be formulated to aid in the achievement of the overall objectives. In addition, early design choices can be made based on the effectiveness of the active controllers.

Appendix J. *EIGENVALUES FOR THE OPTIMUM SATELLITE DESIGN*

1 BEAM DIA = 0.05 M PRIMARY MIRROR DIA = 1.5352 M -- UNITS = SI

OCTOBER 30, 1990 MSC/NASTRAN 10/20/89 PAGE 22

MODIFIED GIVENS METHOD FOR 13 MODES

0

R E A L E I G E N V A L U E S

MODE NO.	EIGENVALUE	RADIANS	CYCLES
1	9.293899E-12 *	3.048590E-06	4.851981E-07
2	-4.288836E-11 *	6.548920E-06	1.042293E-06
3	1.888623E-10 *	1.374272E-05	2.187221E-06
4	3.875300E-10 *	1.968578E-05	3.133089E-06
5	4.043557E-10 *	2.010860E-05	3.200382E-06
6	-7.551080E-10 *	2.747923E-05	4.373455E-06
7	1.542095E+03	3.926952E+01	6.249938E+00
8	1.551844E+03	3.939345E+01	6.269662E+00
9	1.116215E+04	1.056511E+02	1.681489E+01
10	1.322165E+04	1.149854E+02	1.830050E+01
11	1.354728E+04	1.163928E+02	1.852448E+01
12	1.369651E+04	1.170321E+02	1.862624E+01
13	2.161733E+04	1.470283E+02	2.340028E+01
14	2.211353E+04	1.487062E+02	2.366732E+01
15	4.007794E+04	2.001947E+02	3.186198E+01
16	1.031621E+05	3.211885E+02	5.111874E+01

17	1.078325E+05	3.283786E+02	5.226308E+01
18	1.184400E+05	3.441511E+02	5.477334E+01

* The first six modes are the rigid body modes and are not included in the modal analysis bond graph.

Appendix K. *BOND GRAPH TERMINOLOGY*

Because of the extensive use of bond graphs as a modeling technique throughout the project, a brief introduction to the terminology is presented here. Extended descriptions along with physical interpretations and applications can be found in text by Rosenberg and Karnopp(Rosenberg, R. and Karnopp, D., 1983). Only basic definitions are presented here.

> A bond graph is a collection of multiport elements bonded together. In the general sense it is a linear graph whose nodes are multiport elements and whose branches are bonds. (Rosenberg, R. and Karnopp, D., 1972)

Bond graphs model the power and energy relationships throughout a system. Specifically, bond graphs depict power flow between elements of the system through the use of power bonds. Power bonds are depicted as single headed arrows with the direction of the arrow indicating the direction of positive power flow. Power bonds connect to an element at a port and, thus, are connections between pairs of element ports. Ports allow the element to interact with its environment. Six quantities are associated with a given port. The bond *power*, $P(t)$, associated with a bond is defined as the scalar product of bond *effort*, $e(t)$, and bond *flow*, $f(t)$.

$$P(t) = e(t) \cdot f(t) \tag{K.1}$$

Effort and flow are called power variables. The *energy*, $E(t)$ transferred between elements is a function of element *displacement* $q(t)$, (the integral of flow), and *momentum*, $p(t)$ (the integral of effort). Momentum and displacement are referred to as energy variables. Table K.1 summarizes the mechanical and electrical analogies of the six quantities.

Table K.1. Linear constitutive relationships.

Variable Name	Generalized	Mechanical Analog	Electrical Analog
effort	$e(t)$	F, Force	e, voltage
flow	$f(t)$	v, velocity	i, current
momentum	$p = \int e(t)dt$	p, momentum	λ, flux linkage
displacement	$q = \int f(t)dt$	x, displacement	q, charge
power	$p(t) = e(t)f(t)$	$F(t)v(t)$	$e(t)i(t)$
energy (kinetic)	$E(p) = \int f(p)dp$	$\int v(p)dp$	$\int i(\lambda)d\ lambda$
energy (potential)	$E(q) = \int e(q)dq$	$\int F(x)dx$	$\int e(q)dg$

K.1 MULTIPORT ELEMENTS

Multiport elements are the nodes of the graph. There are nine basic multiport elements. They are listed along with their symbol in Figure K.1. They can be grouped according to their energy characteristics as sources, storages, dissipater or junctions.

K.1.1 Sources Sources provide inputs to the system. There are two types of sources: an effort source defined by:

$$e = e(t) \tag{K.2}$$

and a flow source defined by:

$$f = f(t) \tag{K.3}$$

K.1.2 Storage There are two types of storage elements: capacitance,C, and inertance, I. The linear constitutive relationship for capacitance is:

$$e(t) = \frac{q}{C} \tag{K.4}$$

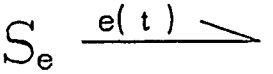
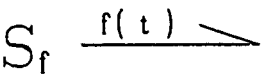
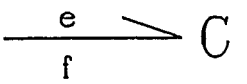
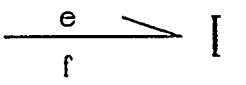
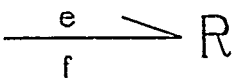
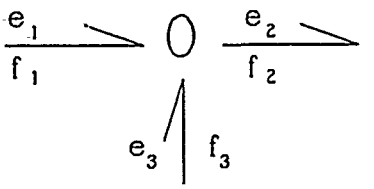
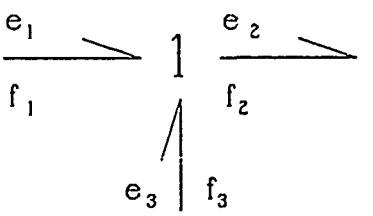
NAME	Symbol
Effort Source	
Flow Source	
Capacitance	
Inertance	
Resistance	
Transformer	
Gyrator	
Common Effort	
Common Flow	

Figure K.1. Basic multiport elements

where,

$$q = \int f(t)dt \tag{K.5}$$

The mechanical and electrical analogs being a spring and capacitor. The linear constitutive relationship for inertance is:

$$f(t) = \frac{p}{I} \tag{K.6}$$

where,

$$p = \int e(t)dt \tag{K.7}$$

The mechanical and electrical analogs being an inertia and inductor.

K.1.3 Dissipater A single resistive, R, element is used to dissipate energy. The linear constitutive relationship for resistance is:

$$e(t) = R \cdot f(t) \tag{K.8}$$

The mechanical and electrical analogs being a damper and resistor.

K.1.4 Junctions Junctions are used to distribute power throughout the system while preserving it (the net power in a junction is zero at all times). There are four junction elements: transformer, gyrator, common effort junction, and common flow junction.

The transformer is a linear 2-port element defined by:

$$e_1 = m \cdot e_2 \tag{K.9}$$

$$f_2 = m \cdot f_1 \tag{K.10}$$

where m is the modulus and e_i is the effort associated with bond i and f_i is the flow associated with bond i.

The gyrator is also a linear 2-port element defined by:

$$e_1 = m \cdot f_2 \tag{K.11}$$

$$e_2 = m \cdot f_1 \tag{K.12}$$

where m is again the modulus.

A common effort junction, also referred to as a 0-junction, is a linear multiport element defined by:

$$e_1 = e_2 = \cdots = e_n \tag{K.13}$$

$$\sum_{1}^{n} f_i = 0 \tag{K.14}$$

A common flow junction, also refered to as a 1-junction, is a linear multiport element defined by:

$$f_1 = f_2 = \cdots = f_n \tag{K.15}$$

$$\sum_{1}^{n} e_i = 0 \tag{K.16}$$

A three port 0 and 1-junctions are shown in Figure K.1

K.2 BONDS

There are two types of bonds: power bonds and signal bonds. Power bonds were described above and are depicted as single headed arrows. In addition to power bond, the situation arises were effort and flow variables are passed between elements without power being transferred. In this case a signal bond is used which is depicted

as a double headed arrow. The variable of interest is indicated on the bond and the other variable is assumed to be zero.

K.3 OVERVIEW

The information presented above is intended only to familiarize the reader with the system modeling technique associated with bond graphs. It presents the basic definitions and symbols of the bond graph language along with formal mathematical equations.

Appendix L. *ENPORT SYSTEM MODEL FILE AND SUPPORTING FORTRAM CODE*

HEADING

FILE MODEL 15:49:22 10/ 9/90 ENPORT-7.3
TRUNI.ENP

TITLE
TOTAL THESIS RUN MODEL

DESCRIPTION
This bond graph is the integrated model of the complete SDI tracking system. It contains the six individual subsystem models integrated together.

SYSTEM GRAPH DESCRIPTION

NODE	TYPE	XLOC	YLOC	ACT	MACRO
SETT	MEGV	-900.	0.	T	
01	M0GV	-700.	0.	T	
TF1	MTGV	-500.	200.	T	
TF2	MTGV	-500.	0.	T	
TF3	MTGV	-500.	-200.	T	
11	M1GV	-300.	400.	T	
12	M1GV	-300.	0.	T	
13	M1GV	-300.	-400.	T	
MTX	MIGV	-300.	500.	T	
MTY	MIGV	-300.	100.	T	
MTZ	MIGV	-300.	-500.	T	
TF4	MTGV	-100.	200.	T	
TF5	MTGV	-100.	0.	T	
TF6	MTGV	-100.	-200.	T	
02	M0GV	100.	0.	T	
SEGT	MEGV	300.	0.	T	
TF7	MTGV	1600.	200.	T	
TF8	MTGV	1600.	0.	T	
TF9	MTGV	1600.	-200.	T	
14	M1GV	1400.	400.	T	
MSX	MIGV	1400.	500.	T	
15	M1GV	1400.	0.	T	
MSY	MIGV	1400.	100.	T	
03	M0GV	1800.	0.	T	
SEGS	MEGV	2000.	0.	T	
16	M1GV	1400.	-400.	T	
MSZ	MIGV	1400.	-500.	T	
ITX	BQGV	300.	400.	T	
ITY	BQGV	300.	200.	T	
ITZ	BQGV	300.	-400.	T	
ISX	BQGV	1200.	300.	T	
ISY	BQGV	1200.	0.	T	
ISZ	BQGV	1200.	-300.	T	
ETI	BAGV	600.	0.	T	
PANG	BAGV	900.	0.	T	
17	M1GV	-300.	-2500.	T	
19	M1GV	400.	-3200.	T	
18	M1GV	-1000.	-3200.	T	
JSX	MIGV	-600.	-3100.	T	
JSY	MIGV	-300.	-2800.	T	
JSZ	MIGV	0.	-3100.	T	
GY1	MGGV	-700.	-2900.	T	
GY2	MGGV	100.	-2900.	T	
GY3	MGGV	-300.	-3200.	T	
A32	BAGV	400.	-2200.	T	
ISRX	BQGV	-1000.	-2200.	T	
ISRZ	BQGV	1200.	-1800.	T	
ISRY	BQGV	800.	-2200.	T	

A30	BAGV	1200.	-1100.	T
A20	BAGV	1200.	-2200.	T
TESTX	B+GV	400.	-800.	T
TESTY	B+GV	600.	-700.	T
TESTZ	B+GV	700.	-500.	T
TFX1	MTGV	4300.	-2100.	T
TFY1	MTGV	4300.	-2300.	T
TFZ1	MTGV	4300.	-2500.	T
TFX2	MTGV	4300.	-2800.	T
TFY2	MTGV	4300.	-3000.	T
TFZ2	MTGV	4300.	-3200.	T
TFX3	MTGV	4300.	-3500.	T
TFY3	MTGV	4300.	-3700.	T
TFZ3	MTGV	4300.	-3900.	T
TFX4	MTGV	4300.	-4200.	T
TFY4	MTGV	4300.	-4400.	T
TFZ4	MTGV	4300.	-4600.	T
TFX5	MTGV	4300.	-4900.	T
TFY5	MTGV	4300.	-5100.	T
TFZ5	MTGV	4300.	-5300.	T
120	M1GV	6400.	-2300.	T
MM1	MIGV	6400.	-2100.	T
RM1	MRGV	6400.	-2600.	T
121	M1GV	6400.	-3000.	T
MM2	MIGV	6400.	-2800.	T
RM2	MRGV	6400.	-3300.	T
122	M1GV	6400.	-3700.	T
MM3	MIGV	6400.	-3500.	T
RM3	MRGV	6400.	-4000.	T
123	M1GV	6400.	-4400.	T
MM4	MIGV	6400.	-4200.	T
RM4	MRGV	6400.	-4700.	T
124	M1GV	6400.	-5100.	T
MM5	MIGV	6400.	-4900.	T
RM5	MRGV	6400.	-5400.	T
CM1	MCGV	6900.	-2300.	T
CM2	MCGV	6900.	-3000.	T
CM3	MCGV	6900.	-3700.	T
CM4	MCGV	6900.	-4400.	T
CM5	MCGV	6900.	-5100.	T
INM1	BQGV	7300.	-1900.	T
INM2	BQGV	7300.	-2600.	T
INM3	BQGV	7300.	-3300.	T
INM4	BQGV	7300.	-4000.	T
INM5	BQGV	7300.	-4700.	T
OPT	BAGV	2500.	-1100.	T
TFXX	MTGV	-1500.	-3200.	T
012	M0GV	-2000.	-3200.	T
SEMWX	MEGV	-2500.	-3200.	T
TFXY	MTGV	-1500.	-3000.	T
TFYY	MTGV	-1500.	-2500.	T
TFYX	MTGV	-1500.	-2700.	T
013	M0GV	-2000.	-2500.	T
SEMWY	MEGV	-2500.	-2500.	T
014	M0GV	800.	-3600.	T
SEMWZ	MEGV	1200.	-4000.	T
TRZ	BAGV	400.	-4000.	T
AZEST	BAGV	3400.	-800.	T
TRPAZ	BAGV	4100.	-600.	T
TRRAZ	BAGV	4100.	-1000.	T
SAZC	BSGV	4700.	-800.	T
ELEST	BAGV	3400.	-1700.	T
TRPEL	BAGV	4100.	-1500.	T
TRREL	BAGV	4100.	-1900.	T
SELC	BSGV	4700.	-1700	T
PATW	BAGV	5100.	-1200.	T

STIX	B-GV	5700.	-800.	T
STIY	B-GV	5700.	-1700.	T
DATA	BAGV	2500.	-1900.	T
SINKD	B-GV	3000.	-2300.	T

CONNECTOR	TYPE	FROM	TO	VERTICES	
B1	BM V	SETT	01		
B2	BM V	01	TF1		
B3	BM V	TF1	11		
B4	BM V	11	MTX		
B5	BM V	11	TF4		
B6	BM V	TF4	02		
B7	BM V	01	TF2		
B8	BM V	TF2	12		
B9	BM V	12	MTY		
B10	BM V	12	TF5		
B11	BM V	TF5	02		
B12	BM V	01	TF3		
B13	BM V	TF3	13		
B14	BM V	13	MTZ		
B15	BM V	13	TF6		
B16	BM V	TF6	02		
B17	BM V	02	SEGT		
B18	BM V	SEGS	03		
B19	BM V	TF7	03		
B20	BM V	TF8	03		
B21	BM V	TF9	03		
B22	BM V	14	TF7		
B23	BM V	14	MSX		
B24	BM V	15	TF8		
B25	BM V	15	MSY		
B27	BM V	16	TF9		
B26	BM V	16	MSZ		
VTX	SM V	11	ITX		
VTY	SM V	12	ITY		
VTZ	SM V	13	ITZ		
VSX	SM V	14	ISX		
VSY	SM V	15	ISY		
VSZ	SM V	16	ISZ		
PTXE	SM V	ITX	ETI		
PTYE	SM V	ITY	ETI		
PTZE	SM V	ITZ	ETI		
PTX	SM V	ETI	PANG	700.	100.
PTY	SM V	ETI	PANG		
PTZ	SM V	ETI	PANG	700.	-100.
PSX	SM V	ISX	PANG		
PSY	SM V	ISY	PANG		
PSZ	SM V	ISZ	PANG		
B34	BR V	18	JSX		
B35	BR V	17	JSY		
B36	BR V	19	JSZ		
WSXB	SR V	18	ISRX		
TSX	SR V	ISRX	A32		
WSYB	SR V	17	A32		
WSZB	SR V	19	A32		
WSY1	SR V	A32	ISRY		
STX	SM V	PANG	A30	900.	-1000.
STY	SM V	PANG	A30	800.	-1100.
STZ	SM V	PANG	A30	700.	-1200.
TSZ	SR V	ISRZ	A30		
WSX1	SR V	A32	A20	800.	-2000.
TSY	SR V	ISRY	A20		
WSZ1	SR V	A32	A20	800.	-2400.
WSX2	SR V	A20	A30	1000.	-1800.
WSY2	SR V	A20	A30	1400.	-1800.

WSZ2	SR V	A20	ISRZ				
B28	BR V	GY1	18				
B29	BR V	17	GY1				
B30	BR V	GY2	17				
B31	BR V	19	GY2				
B32	BR V	GY3	19				
B33	BR V	18	GY3				
0X	SM V	TESTX	ETI				
0Y	SM V	TESTY	ETI				
0Z	SM V	TESTZ	ETI				
B71	BM V	120	MM1				
B73	BM V	120	RM1				
B78	BM V	121	MM2				
B80	BM V	121	RM2				
B85	BM V	122	MM3				
B87	BM V	122	RM3				
B92	BM V	123	MM4				
B94	BM V	123	RM4				
B99	BM V	124	MM5				
B101	BM V	124	RM5				
B72	BM V	120	CM1				
B79	BM V	121	CM2				
B86	BM V	122	CM3				
B93	BM V	123	CM4				
B100	BM V	124	CM5				
NUD1	SM V	120	INM1				
NUD2	SM V	121	INM2				
NUD3	SM V	122	INM3				
NUD4	SM V	123	INM4				
NUD5	SM V	124	INM5				
STMX	SM V	A30	OPT	1800.	-800.		
STMY	SM V	A30	OPT				
STMZ	SM V	A30	OPT	1800.	-1400.		
NU1	SM V	INM1	OPT	8000.	-1900.	8000.	-300.
				2900.	-300.		
NU2	SM V	INM2	OPT	8100.	-2600.	8100.	-200.
				2800.	-200.		
NU3	SM V	INM3	OPT	8200.	-3300.	8200.	-100.
				2700.	-100.		
NU4	SM V	INM4	OPT	8300.	-4000.	8300.	0.
				2600.	0.		
NU5	SM V	INM5	OPT	8400.	-4700.	8400.	100.
				2500.	100.		
B37	BR V	SEMWX	012				
B38	BR V	012	TFXX				
B39	BR V	TFXX	18				
B40	BR V	012	TFXY				
B41	BR V	TFXY	17				
B47	BR V	SEMWY	013				
B48	BR V	013	TFYY				
B49	BR V	TFYY	17				
B50	BR V	013	TFYX				
B51	BR V	TFYX	18				
B42	BR V	SEMWZ	014				
B43	BR V	014	19				
B52	BM V	012	TFX1	-400.	-4400.	1500.	-4400.
B53	BM V	012	TFX2	-500.	-4500.	1500.	-4500.
B54	BM V	012	TFX3	-600.	-4600.	1500.	-4600.
B55	BM V	012	TFX4	-700.	-4700.	1500.	-4700.
B56	BM V	012	TFX5	-800.	-4800.	1500.	-4800.
B57	BM V	013	TFY1	-2300.	-2100.	-2800.	-2100.
				-2800.	-5100.	1500.	-5100.
B58	BM V	013	TFY2	-2200.	-2000.	-2900.	-2000.
				-2900.	-5200.	1500.	-5200.
B59	BM V	013	TFY3	-2100.	-1900.	-3000.	-1900.
				-3000.	-5300.	1500.	-5300.

B60	BM V	013	TFY4	-2000.	-1800.	-3100.	-1800.
				-3100.	-5400.	1500.	-5400.
B61	BM V	013	TFY5	-1900.	-1700.	-3200.	-1700.
				-3200.	-5500.	1500.	-5500.
B62	BM V	014	TFZ1	1500.	-3200.		
B63	BM V	014	TFZ2	1500.	-3300.		
B64	BM V	014	TFZ3	1500.	-3400.		
B65	BM V	014	TFZ4	1500.	-3500.		
B66	BM V	014	TFZ5	1500.	-3600.		
WSZB2	SR V	19	TRZ				
TIZ	SR V	TRZ	SEMWZ				
B67	BM V	TFX1	120				
B68	BM V	TFY1	120				
B69	BM V	TFZ1	120				
B74	BM V	TFX2	121				
B75	BM V	TFY2	121				
B76	BM V	TFZ2	121				
B81	BM V	TFX3	122				
B82	BM V	TFY3	122				
B83	BM V	TFZ3	122				
B88	BM V	TFX4	123				
B89	BM V	TFY4	123				
B90	BM V	TFZ4	123				
B95	BM V	TFX5	124				
B96	BM V	TFY5	124				
B97	BM V	TFZ5	124				
XBAR	SM V	OPT	AZEST				
ZBAR	SM V	OPT	ELEST				
AZPER	SM V	AZEST	TRPAZ				
AZRER	SM V	AZEST	TRRAZ				
AZPC	SM V	TRPAZ	SAZC				
AZRC	SM V	TRRAZ	SAZC				
AZC	SM V	SAZC	PATW				
ELPER	SM V	ELEST	TRPEL				
ELRER	SM V	ELEST	TRREL				
ELPC	SM V	TRPEL	SELC				
ELRC	SM V	TRREL	SELC				
ELC	SM V	SELC	PATW				
TIX	SM V	PATW	STIX				
TIY	SM V	PATW	STIY				
MAZPER	SM V	OPT	DATA	2100.	-1700.		
MELPER	SM V	OPT	DATA	2300.	-1700.		
STMMX	SM V	OPT	DATA				
STMMY	SM V	OPT	DATA	2700.	-1700.		
STMMZ	SM V	OPT	DATA	2900.	-1700.		
DOJ	SM V	DATA	SINKD				

GRAPHICAL ENVIRONMENT

Node size:	4.0000E-01
Connector size:	6.2500E-01
Scale factor:	1.0000E+00
Horizontal window minimum:	-3.2500E+03
Horizontal window maximum:	8.4500E+03
Vertical window minimum:	-5.8626E+03
Vertical window maximum:	1.4626E+03
Framing enabled:	ON
Gridding visible:	ON
Grid size:	1.0000E+02
Default node regime enabled:	ON
Default connector regime enabled:	OFF
Default node regime:	G
Default connector regime:	G
Postprocessor grid:	ON
Postprocessor box:	ON

```
Color 1    0   0   0 Black       (background)
Color 2    0 100   0 White       (default line and label color)
Color 3  120  50 100 Red         (attention)
Color 4  240  50 100 Green       (Mechanical)
Color 5    0  50  80 Blue        (Rotational)
Color 6  300  50 100 Cyan        (Electrical)
Color 7   60  50 100 Magenta     (Hydraulic)
Color 8  180  50 100 Yellow      (Thermal)
```

NODE EQUATIONS

Named_parameters: 34

1	JSXP	1.642000E-03
2	JSYP	1.642000E-03
3	JSZP	1.186380E-02
4	RM1P	6.233478E+00
5	RM2P	6.267690E+00
6	RM3P	1.587265E+01
7	RM4P	1.912553E+01
8	RM5P	1.921330E+01
9	CM1P	1.079300E+04
10	CM2P	1.091200E+04
11	CM3P	6.998300E+04
12	CM4P	1.016070E+05
13	CM5P	1.025410E+05
14	AXP	0.000000E+00
15	AYP	0.000000E+00
16	AZP	0.000000E+00
17	AEPD1P	2.200000E+00
18	AEPD2P	1.000000E+00
19	AEPN1P	2.141700E+03
20	AEPN2P	9.423480E+03
21	TRZD1P	0.000000E+00
22	TRZD2P	1.000000E+02
23	TRZD3P	1.000000E+00
24	TRZN1P	2.115232E+06
25	TRZN2P	2.644040E+05
26	TRZN3P	0.000000E+00
27	AERD1P	1.000000E+02
28	AERD2P	1.000000E+00
29	AERN1P	1.892945E+07
30	AERN2P	2.366181E+06
31	MCP	0.000000E+00
32	JSXPI	-6.090000E+02
33	JSYPI	-6.090000E+02
34	JSZPI	-8.400000E+01

Number of outputs: 100

```
Node:  SETT        Connectors:  B1
  Equation:   Y = ZZSU04  ( X, P )      1  1  1
    Y_list          X_list           Parameters  Index
    E.B1            TIME             1.0000E+00    0

Node:  TF1         Connectors:  B2         B3
       F.B2         = MOD.TF1      * F.B3
       E.B3         = MOD.TF1      * E.B2
  Equation:   Y = ZZSU02  ( X, P )      1  3  1
    Y_list          X_list           Parameters  Index
    MOD.TF1         VTX              1.0000E+00    0
                    VTY
                    VTZ
```

```
Node:  TF2          Connectors:  B7          B8
       F.B7          = MOD.TF2       * F.B8
       E.B8          = MOD.TF2       * E.B7
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF2          VTX                 2.0000E+00    0
                     VTY
                     VTZ

Node:  TF3          Connectors:  B12         B13
       F.B12         = MOD.TF3       * F.B13
       E.B13         = MOD.TF3       * E.B12
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF3          VTX                 3.0000E+00    0
                     VTY
                     VTZ

Node:  MTX          Connectors:  B4
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B4             P.B4                1.0000E+00    0

Node:  MTY          Connectors:  B9
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B9             P.B9                1.0000E+00    0

Node:  MTZ          Connectors:  B14
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B14            P.B14               1.0000E+00    0

Node:  TF4          Connectors:  B5          B6
       E.B5          = MOD.TF4       * E.B6
       F.B6          = MOD.TF4       * F.B5
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF4          PTXE                1.0000E+00    0
                     PTYE
                     PTZE

Node:  TF5          Connectors:  B10         B11
       E.B10         = MOD.TF5       * E.B11
       F.B11         = MOD.TF5       * F.B10
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF5          PTXE                2.0000E+00    0
                     PTYE
                     PTZE

Node:  TF6          Connectors:  B15         B16
       E.B15         = MOD.TF6       * E.B16
       F.B16         = MOD.TF6       * F.B15
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF6          PTXE                3.0000E+00    0
                     PTYE
                     PTZE

Node:  SEGT         Connectors:  B17
  Equation:    Y = ZZSU03   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    E.B17            PTXE                1.0000E+00    0
                     PTYE
                     PTZE
```

```
Node:  TF7          Connectors:  B19        B22
       F.B19         = MOD.TF7       * F.B22
       E.B22         = MOD.TF7       * E.B19
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF7          PSX                 1.0000E+00    0
                     PSY
                     PSZ

Node:  TF8          Connectors:  B20        B24
       F.B20         = MOD.TF8       * F.B24
       E.B24         = MOD.TF8       * E.B20
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF8          PSX                 2.0000E+00    0
                     PSY
                     PSZ

Node:  TF9          Connectors:  B21        B27
       F.B21         = MOD.TF9       * F.B27
       E.B27         = MOD.TF9       * E.B21
  Equation:    Y = ZZSU02   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    MOD.TF9          PSX                 3.0000E+00    0
                     PSY
                     PSZ

Node:  MSX          Connectors:  B23
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B23            P.B23               1.0000E+00    0

Node:  MSY          Connectors:  B25
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B25            P.B25               1.0000E+00    0

Node:  SEGS         Connectors:  B18
  Equation:    Y = ZZSU03   ( X, P )       1  3  1
    Y_list           X_list              Parameters  Index
    E.B18            PSX                 1.0000E+00    0
                     PSY
                     PSZ

Node:  MSZ          Connectors:  B26
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B26            P.B26               1.0000E+00    0

Node:  ETI          Connectors:  PTXE       PTYE       PTZE
  Equation:    Y = ZZSU07   ( X, P )       3  4  1
    Y_list           X_list              Parameters  Index
    PTX              PTXE                1.0000E+00    0
    PTY              PTYE
    PTZ              PTZE
                     TIME

Node:  PANG         Connectors:  PTX        PTY        PTZ
  Equation:    Y = ZZSU08   ( X, P )       3  6  1
    Y_list           X_list              Parameters  Index
    STX              PTX                 1.0000E+00    0
    STY              PTY
    STZ              PTZ
                     PSX
                     PSY
```

```
Node:  JSX          Connectors:  B34
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B34            P.B34               1.6420E-03    1

Node:  JSY          Connectors:  B35
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B35            P.B35               1.6420E-03    2

Node:  JSZ          Connectors:  B36
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    F.B36            P.B36               1.1864E-02    3

Node:  GY1          Connectors:  B28         B29
       E.B28          = MOD.GY1       * F.B29
       E.B29          = MOD.GY1       * F.B28
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    MOD.GY1          WSZB               -8.4000E+01   34

Node:  GY2          Connectors:  B30         B31
       E.B30          = MOD.GY2       * F.B31
       E.B31          = MOD.GY2       * F.B30
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    MOD.GY2          WSXB               -6.0900E+02   32

Node:  GY3          Connectors:  B32         B33
       E.B32          = MOD.GY3       * F.B33
       E.B33          = MOD.GY3       * F.B32
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    MOD.GY3          WSYB               -6.0900E+02   33

Node:  A32          Connectors:  TSX         WSYB        WSZB
  Equation:    Y = ZZSU09   ( X, P )       3  4  1
    Y_list           X_list              Parameters  Index
    WSX1             WSXB                1.0000E+00    0
    WSY1             WSYB
    WSZ1             WSZB
                     TSX

Node:  A30          Connectors:  STX         STY         STZ
  Equation:    Y = ZZSU11   ( X, P )       3  6  1
    Y_list           X_list              Parameters  Index
    STMX             STX                 1.0000E+00    0
    STMY             STY
    STMZ             STZ
                     TSX
                     TSY
                     TSZ

Node:  A20          Connectors:  WSX1        TSY         WSZ1
  Equation:    Y = ZZSU10   ( X, P )       3  5  1
    Y_list           X_list              Parameters  .ndex
    WSX2             WSX1                1.0000E+00    0
    WSY2             WSY1
    WSZ2             WSZ1
                     TSY
                     TSZ

Node:  TESTX        Connectors:  0X
```

```
  Equation:   Y = CON      ( X, P )      1  0  1
    Y_list          X_list              Parameters  Index
    0X                                  0.0000E+00    0

Node:  TESTY      Connectors:  0Y
  Equation:   Y = CON      ( X, P )      1  0  1
    Y_list          X_list              Parameters  Index
    0Y                                  0.0000E+00    0

Node:  TESTZ      Connectors:  0Z
  Equation:   Y = CON      ( X, P )      1  0  1
    Y_list          X_list              Parameters  Index
    0Z                                  0.0000E+00    0

Node:  TFX1       Connectors:  B52       B67
       F.B52       = MOD.TFX1     * F.B67
       E.B67       = MOD.TFX1     * E.B52
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFX1                            6.4000E+01    0
                                        1.0000E+00    0

Node:  TFY1       Connectors:  B57       B68
       F.B57       = MOD.TFY1     * F.B68
       E.B68       = MOD.TFY1     * E.B57
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFY1                            1.8500E+02    0
                                        1.0000E+00    0

Node:  TFZ1       Connectors:  B62       B69
       F.B62       = MOD.TFZ1     * F.B69
       E.B69       = MOD.TFZ1     * E.B62
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFZ1                            1.2600E+02    0
                                        1.0000E+00    0

Node:  TFX2       Connectors:  B53       B74
       F.B53       = MOD.TFX2     * F.B74
       E.B74       = MOD.TFX2     * E.B53
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFX2                            6.4000E+01    0
                                        2.0000E+00    0

Node:  TFY2       Connectors:  B58       B75
       F.B58       = MOD.TFY2     * F.B75
       E.B75       = MOD.TFY2     * E.B58
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFY2                            1.8500E+02    0
                                        2.0000E+00    0

Node:  TFZ2       Connectors:  B63       B76
       F.B63       = MOD.TFZ2     * F.B76
       E.B76       = MOD.TFZ2     * E.B63
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
    Y_list          X_list              Parameters  Index
    MOD.TFZ2                            1.2600E+02    0
                                        2.0000E+00    0

Node:  TFX3       Connectors:  B54       B81
       F.B54       = MOD.TFX3     * F.B81
       E.B81       = MOD.TFX3     * E.B54
  Equation:   Y = ZZSU17   ( X, P )      1  0  2
```

```
    Y_list          X_list          Parameters  Index
    MOD.TFX3                        6.4000E+01    0
                                    3.0000E+00    0

Node:  TFY3       Connectors:  B59       B82
       F.B59       = MOD.TFY3     * F.B82
       E.B82       = MOD.TFY3     * E.B59
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFY3                        1.8500E+02    0
                                    3.0000E+00    0

Node:  TFZ3       Connectors:  B64       B83
       F.B64       = MOD.TFZ3     * F.B83
       E.B83       = MOD.TFZ3     * E.B64
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFZ3                        1.2600E+02    0
                                    3.0000E+00    0

Node:  TFX4       Connectors:  B55       B88
       F.B55       = MOD.TFX4     * F.B88
       E.B88       = MOD.TFX4     * E.B55
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFX4                        6.4000E+01    0
                                    4.0000E+00    0

Node:  TFY4       Connectors:  B60       B89
       F.B60       = MOD.TFY4     * F.B89
       E.B89       = MOD.TFY4     * E.B60
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFY4                        1.8500E+02    0
                                    4.0000E+00    0

Node:  TFZ4       Connectors:  B65       B90
       F.B65       = MOD.TFZ4     * F.B90
       E.B90       = MOD.TFZ4     * E.B65
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFZ4                        1.2600E+02    0
                                    4.0000E+00    0

Node:  TFX5       Connectors:  B56       B95
       F.B56       = MOD.TFX5     * F.B95
       E.B95       = MOD.TFX5     * E.B56
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFX5                        6.4000E+01    0
                                    5.0000E+00    0

Node:  TFY5       Connectors:  B61       B96
       F.B61       = MOD.TFY5     * F.B96
       E.B96       = MOD.TFY5     * E.B61
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFY5                        1.8500E+02    0
                                    5.0000E+00    0

Node:  TFZ5       Connectors:  B66       B97
       F.B66       = MOD.TFZ5     * F.B97
       E.B97       = MOD.TFZ5     * E.B66
  Equation:   Y = ZZSU17  ( X, P )      1  0  2
    Y_list          X_list          Parameters  Index
    MOD.TFZ5                        1.2600E+02    0
```

```
                                        5.0000E+00    0

Node:  MM1          Connectors:  B71
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    F.B71            P.B71               1.0000E+00    0

Node:  RM1          Connectors:  B73
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B73            F.B73               6.2335E+00    4

Node:  MM2          Connectors:  B78
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    F.B78            P.B78               1.0000E+00    0

Node:  RM2          Connectors:  B80
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B80            F.B80               6.2677E+00    5

Node:  MM3          Connectors:  B85
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    F.B85            P.B85               1.0000E+00    0

Node:  RM3          Connectors:  B87
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B87            F.B87               1.5873E+01    6

Node:  MM4          Connectors:  B92
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    F.B92            P.B92               1.0000E+00    0

Node:  RM4          Connectors:  B94
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B94            F.B94               1.9126E+01    7

Node:  MM5          Connectors:  B99
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    F.B99            P.B99               1.0000E+00    0

Node:  RM5          Connectors:  B101
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B101           F.B101              1.9213E+01    8

Node:  CM1          Connectors:  B72
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B72            Q.B72               1.0793E+04    9

Node:  CM2          Connectors:  B79
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B79            Q.B79               1.0912E+04   10

Node:  CM3          Connectors:  B86
  Equation:    Y = GAIN     ( X, P )      1  1  1
    Y_list           X_list              Parameters  Index
    E.B86            Q.B86               6.9983E+04   11
```

```
Node:  CM4          Connectors:  B93
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    E.B93            Q.B93               1.0161E+05   12

Node:  CM5          Connectors:  B100
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    E.B100           Q.B100              1.0254E+05   13

Node:  OPT          Connectors:  STMX       STMY       STMZ
  Equation:    Y = ZZSU12   ( X, P )       7  9  4
    Y_list           X_list              Parameters  Index
    XBAR             STMX                0.0000E+00   14
    ZBAR             STMY                0.0000E+00   15
    MAZPER           STMZ                0.0000E+00   16
    MELPER           TIME                0.0000E+00   31
    STMMX            NU1
    STMMY            NU2
    STMMZ            NU3
                     NU4
                     NU5

Node:  TFXX         Connectors:  B38        B39
       F.B38          = MOD.TFXX      * F.B39
       E.B39          = MOD.TFXX      * E.B38
  Equation:    Y = CON      ( X, P )       1  0  1
    Y_list           X_list              Parameters  Index
    MOD.TFXX                             1.0000E+00    0

Node:  SEMWX        Connectors:  B37
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    E.B37            TIX                 1.0000E+00    0

Node:  TFXY         Connectors:  B40        B41
       F.B40          = MOD.TFXY      * F.B41
       E.B41          = MOD.TFXY      * E.B40
  Equation:    Y = CON      ( X, P )       1  0  1
    Y_list           X_list              Parameters  Index
    MOD.TFXY                             0.0000E+00    0

Node:  TFYY         Connectors:  B48        B49
       F.B48          = MOD.TFYY      * F.B49
       E.B49          = MOD.TFYY      * E.B48
  Equation:    Y = CON      ( X, P )       1  0  1
    Y_list           X_list              Parameters  Index
    MOD.TFYY                             1.0000E+00    0

Node:  TFYX         Connectors:  B50        B51
       F.B50          = MOD.TFYX      * F.B51
       E.B51          = MOD.TFYX      * E.B50
  Equation:    Y = CON      ( X, P )       1  0  1
    Y_list           X_list              Parameters  Index
    MOD.TFYX                             0.0000E+00    0

Node:  SEMWY        Connectors:  B47
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    E.B47            TIY                 1.0000E+00    0

Node:  SEMWZ        Connectors:  B42        TIZ
  Equation:    Y = GAIN     ( X, P )       1  1  1
    Y_list           X_list              Parameters  Index
    E.B42            TIZ                -1.0000E+00    0
```

```
Node:  TRZ          Connectors:  WSZB2      TIZ
  Equation:    Y = TRANSFER( X,  P )        1  1  8
    Y_list            X_list              Parameters  Index
    TIZ               WSZB2               0.0000E+00    0
                                          0.0000E+00    0
                                          9.4460E+00    0
                                          1.0000E+00    0
                                          6.6218E+03    0
                                          1.8727E+04    0
                                          9.1977E+02    0
                                          0.0000E+00    0

Node:  AZEST        Connectors:  XBAR       AZPER      AZRER
  Equation:    Y = ZZSU15   ( X,  P )        2  2  1
    Y_list            X_list              Parameters  Index
    AZPER             XBAR                1.0000E+00    0
    AZRER             TIME

Node:  TRPAZ        Connectors:  AZPER      AZPC
  Equation:    Y = TRANSFER( X,  P )        1  1  6
    Y_list            X_list              Parameters  Index
    AZPC              AZPER               0.0000E+00    0
                                          9.4460E+00    0
                                          1.0000E+00    0
                                          6.6544E+03    0
                                          1.3530E+05    0
                                          4.7843E+04    0

Node:  TRRAZ        Connectors:  AZRER      AZRC
  Equation:    Y = GAIN     ( X,  P )        1  1  1
    Y_list            X_list              Parameters  Index
    AZRC              AZRER               0.0000E+00    0

Node:  SAZC         Connectors:  AZPC       AZRC       AZC
  Equation:    Y = SUM      ( X,  P )        1  2  2
    Y_list            X_list              Parameters  Index
    AZC               AZPC                1.0000E+00    0
                      AZRC                1.0000E+00    0

Node:  ELEST        Connectors:  ZBAR       ELPER      ELRER
  Equation:    Y = ZZSU16   ( X,  P )        2  2  1
    Y_list            X_list              Parameters  Index
    ELPER             ZBAR                1.0000E+00    0
    ELRER             TIME

Node:  TRPEL        Connectors:  ELPER      ELPC
  Equation:    Y = TRANSFER( X,  P )        1  1  6
    Y_list            X_list              Parameters  Index
    ELPC              ELPER               0.0000E+00    0
                                          9.4460E+00    0
                                          1.0000E+00    0
                                          6.6544E+03    0
                                          1.3530E+05    0
                                          4.7843E+04    0

Node:  TRREL        Connectors:  ELRER      ELRC
  Equation:    Y = GAIN     ( X,  P )        1  1  1
    Y_list            X_list              Parameters  Index
    ELRC              ELRER               0.0000E+00    0

Node:  SELC         Connectors:  ELPC       ELRC       ELC
  Equation:    Y = SUM      ( X,  P )        1  2  2
    Y_list            X_list              Parameters  Index
    ELC               ELPC               -1.0000E+00    0
                      ELRC               -1.0000E+00    0
```

```
Node:  PATW        Connectors:  AZC        ELC        TIX
  Equation:    Y = MATRIX  ( X, P )       2  2  0

                      AZC           ELC
      TIX             0.0000E+00   -1.0000E+00
      TIY             1.0000E+00    0.0000E+00

Node:  DATA        Connectors:  MAZPER     MELPER     STMMX
  Equation:    Y = ZZSU18  ( X, P )       1  6  1
    Y_list             X_list            Parameters  Index
    DOJ                MAZPER            1.0000E+00    0
                       MELPER
                       STMMX
                       STMMY
                       STMMZ
                       TIME
```

INITIAL CONDITIONS

P.B4	=	2.7918E+01
P.B9	=	4.1480E+01
P.B14	=	0.0000E+00
P.B23	=	0.0000E+00
P.B25	=	0.0000E+00
P.B26	=	0.0000E+00
PTXE	=	3.5626E+06
PTYE	=	5.2818E+06
PTZE	=	0.0000E+00
PSX	=	0.0000E+00
PSY	=	0.0000E+00
PSZ	=	0.0000E+00
P.B34	=	0.0000E+00
P.B35	=	0.0000E+00
P.B36	=	0.0000E+00
TSX	=	0.0000E+00
TSZ	=	0.0000E+00
TSY	=	-3.1416E+00
P.B71	=	0.0000E+00
P.B78	=	0.0000E+00
P.B85	=	0.0000E+00
P.B92	=	0.0000E+00
P.B99	=	0.0000E+00
Q.B72	=	0.0000E+00
Q.B79	=	0.0000E+00
Q.B86	=	0.0000E+00
Q.B93	=	0.0000E+00
Q.B100	=	0.0000E+00
NU1	=	0.0000E+00
NU2	=	0.0000E+00
NU3	=	0.0000E+00
NU4	=	0.0000E+00
NU5	=	0.0000E+00
XTF01	=	0.0000E+00
XTF02	=	0.0000E+00
XTF03	=	0.0000E+00
XTF04	=	0.0000E+00
XTF05	=	0.0000E+00
XTF06	=	0.0000E+00
XTF07	=	0.0000E+00

OUTPUT VARIABLES

1
P.B4

TIME CONTROLS

Initial time = 0.0000E+00
Final time = 3.6300E+02
Number saved = 512

END-FILE

```
CFILE:FCNZZS ------------------ FCNZZS -------------------------------
C               user-defined vector functions ZZSU01 ... ZZSU99.
C
C---  DESCRIPTION: These subroutines are called by GETVL in SOLVE1
C                  for evaluation. They are called by FTEXT1 in FCNLIB
C                  for description of their definitions.
C
C---  CONTENTS:  ZZSU01   user-defined function 1
C                ZZSU02
C                  .
C                  .
C                ZZSU98
C                ZZSU99
C
C     NOTE.   Each subroutine has the same calling structure, namely,
C                CALL ZZSUij ( X,P,Y,DOF,NUML,DESC)
C
C             X are the values of the input arguments;
C             P are the values of the input parameters; and
C             Y are the values of the output arguments returned by SUBRi.
C             DOF=.T., return the description; else, evaluate.
C             NUML is the number of lines in the description.
C             DESC are the lines of description.
C
C             X,P,Y are dimensioned in subroutine GETVL1.
C             1) Prepare function description as reminder to user.
C                a)  Set NUML to the number of lines of the description.
C                    The maximum value is 20.
C                b)  Set DESC(1) to the title.
C                c)  Set DESC(2-NUML) to any additional description.
C             2) Prepare FORTRAN evaluation code and insert it after
C                the proper comment line C in the subroutine.
C
C---- WARNING!   Be sure NUML is set to 0 if ZZSUij is not in use.
C                Otherwise an errant request to see it could cause a
C                fatal FORTRAN error.
C
C---  Last revision:  April 28, 1989.
C
CEOFH:FCNZZS ------------------------------------------------------------
C>>>>>
C
C
CZZSU01 >>>>>>>>>>>>>>>>>>> ZZSU01 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU01(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:  Use as a single-output-variable function.
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of the function
C             NUML,   number of lines in the description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
```



```
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU01*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 1
        DESC(1)=' ZZSU01: NOT USED'
C
      ELSE
C----- Evaluation section
      Y(1)=P(1)*X(1)
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU02 >>>>>>>>>>>>>>>>>>> ZZSU02 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU02(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU02*********************************************************************
C
C---  For detailed instructions see file header.

      IF (DOF) THEN
C
C----- Description section (max 20 lines)
        NUML= 4
        DESC(1)=' ZZSU02: TRANSFORMER MODULATION'
        DESC(2)=' CALCULATES THE MODULUS OF TRANSFORMERS THAT ARE'
        DESC(3)=' THAT ARE FUNCTIONS OF THE DIRECTION OF A VECTOR'
        DESC(4)=' THIS FUNCTION WORKS LIKE A COSINE FUNCTION.'
C
      ELSE
C----- Evaluation section
C
C Y(1): MODULATOR OF TRANSFORMER
C X(1): VECTOR COMPONENT #1
C X(2): VECTOR COMPONENT #2
C X(3): VECTOR COMPONENT #3
C P(1): DIRECTION OF INTEREST
C
```

```
      Y(1)=X(INT(P(3)))/DSQRT(X(1)**2+X(2)**2+X(3)**2)
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU03 >>>>>>>>>>>>>>>>>>> ZZSU03 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU03(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of the function
C             NUML,   number of lines in the description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU03**********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 3
        DESC(1)=' ZZSU03: CALCULATES THE ACCELERATION OF AN ORBITING'
        DESC(2)=' BODY DUE TO GRAVITY.  INPUT IS THE POSITION OF THE'
        DESC(3)=' BODY IN THE EARTH FRAME.'
C
      ELSE
C----- Evaluation section
C
C X(1): X POSITION OF BODY
C X(2): Y POSITION OF BODY
C X(3): Z POSITION OF BODY
C Y(1): ACCELERATION OF THE BODY DUE TO GRAVITY
C XMU: EARTH'S GRAVITAIONAL CONSTANT
C
      XMU=3.986012D+14
C
      Y(1)=XMU/(X(1)**2+X(2)**2+X(3)**2)
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU04 >>>>>>>>>>>>>>>>>>> ZZSU04 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU04(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
```



```
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z).
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU04**********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 2
      DESC(1)='ZZSU04: CALCULATES THE ACCELERATION OF THE MISSILE'
      DESC(2)='DUE TO THE ROCKET THRUST.  THIS IS A TIME DEPENDENT'
      DESC(3)='THAT CHANGES AS THE MISSILE STAGES BURN OUT.'
C
      ELSE
C----- Evaluation section
C
C X(1): TIME
C Y(1): THRUST OUTPUT BY ROCKET
C THRUST: THRUST OF ROCKET STAGE
C XISP: SPECIFIC IMPULSE STAGE
C TOTALM: TOTAL MASS OF ROCKET
C G:  ACCLERATION OF GRAVITY
C XLBFNWT: FACTOR FOR CONVERSION FROM POUNDS FORCE TO NEWTONS
C XLBMKG: FACTOR FOR CONVERSION FROM POUNDS MASS TO KILOGRAMS
C
      G=9.806194D0
      XLBFNWT=4.448444D0
      XLBMKG=0.45359237D0
C
C FIRST STAGE
C
      IF(X(1) .LE. 151.7462)THEN
      THRUST=464900.0D0
      ISP=256.0D0
      TOTALM=382871.0D0
      STIME=0.0D0
C
C SECOND STAGE
C
      ELSEIF(X(1) .GT. 151.7462D0 .AND. X(1) .LE. 360.5306D0)THEN
      THRUST=102300.0D0
      ISP=317.4D0
      TOTALM=77551.0D0
      STIME=151.7462D0
C
C FREEFALL
C
      ELSE
      THRUST=0.0D0
      ISP=100.0D0
      TOTALM=4000.0D0
      STIME=360.5306D0
```

```
      ENDIF
C
C CALCULATE ACCELERATION
C
      Y(1)=(THRUST*XLBFNWT)/(TOTALM*XLBMKG-THRUST*XLBFNWT*
     +(X(1)-STIME)/(ISP*G))
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU05 >>>>>>>>>>>>>>>>>>> ZZSU05 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU05(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),F(40),U(40,5),UT(5,40)
C
C***ZZSU05*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 1
          DESC(1)='ZZSU05: NOT USED'
C
      ELSE
      Y(1)=X(1)*P(1)
C----- Evaluation section
C
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU06 >>>>>>>>>>>>>>>>>>> ZZSU06 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU06(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
```



```
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),DIS(40),U(40,5)
C
C***ZZSU06*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 1
C
      ELSE
C----- Evaluation section
C
      Y(1)=X(1)*P(1)
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU07 >>>>>>>>>>>>>>>>>>> ZZSU07 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU07(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),A(3,3)
C
C***ZZSU07*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lin
        NUML= 3
        DESC(1)=' ZZSU07: CONVERTS ROTATING EARTH FRAME POSITION VECTOR'
        DESC(2)=' INTO INERTIAL FRAME POSITION VECTOR.  THIS IS THE '
        DESC(3)=' SUBROUTINE IN FUNCTION BLOCK ETI.'
C
      ELSE
```

```
C----- Evaluation section
C
C X(1): INERTIAL FRAME X COORDINATE
C X(2): INERTIAL FRAME Y COORDINATE
C X(3): INERTIAL FRAME Z COORDINATE
C X(4): TIME(SEC)
C Y(1): EARTH FRAME X COORDINATE
C Y(2): EARTH FRAME Y COORDINATE
C Y(3): EARTH FRAME Z COORDINATE
C EOMEGA: EARTH ROTATION RATE(RADS/SEC)
C ETHETA: EARTH ROTATION ANGLE(RADS)
C A: ROTATION MATRIX FROM EARTH TO INERTIAL COORDINATES
C
      EOMEGA=2.0D0*3.14159265D0/86164.09D0
      ETHETA=X(4)*EOMEGA
C
      A(1,1)=DCOS(ETHETA)
      A(1,2)=0.0D0
      A(1,3)=DSIN(ETHETA)
      A(2,3)=0.0D0
      A(2,2)=1.0D0
      A(2,3)=0.0D0
      A(3,1)=-1.0D0*DSIN(ETHETA)
      A(3,2)=0.0D0
      A(3,3)=DCOS(ETHETA)
      DO 10 I=1,3
      Y(I)=0.0D0
      DO 20 J=1,3
      Y(I)=Y(I)+X(J)*A(I,J)
 20   CONTINUE
 10   CONTINUE
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU08 >>>>>>>>>>>>>>>>>>> ZZSU08<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU08(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU08**********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
```



```
        NUML= 2
        DESC(1)=' ZZSU08: CALCULATES A VECTOR FROM THE SYSTEM TO THE'
        DESC(2)=' TARGET BY PERFORMING VECTOR SUBSTRACTION.  THIS '
        DESC(3)=' SUBROUTINE IS USED IN THE FUNCTION BLOCK PANG.'
C
      ELSE
C----- Evaluation section
C
C X(1): TARGET X POSITION
C X(2): TARGET Y POSITION
C X(3): TARGET Z POSITION
C X(4): SYSTEM X POSITION
C X(5): SYSTEM Y POSITION
C X(6): SYSTEM Z POSITION
C
      DO 10 I=1,3
      Y(I)=X(I)-X(I+3)
 10   CONTINUE
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU09 >>>>>>>>>>>>>>>>>>> ZZSU09 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU09(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),A(3,3)
C
C***ZZSU09**********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 2
        DESC(1)=' ZZSU09: TRANSFORMS THE RIGID BODY ROTATION RATES THROUGH ONE'
        DESC(2)=' ANGLE ABOUT THE X-AXIS.  USED IN FUNCTION BLOCK'
        DESC(3)=' A32.'
C
      ELSE
C----- Evaluation section
C
C X(1): ROTATION RATE ABOUT X AXIS
C X(2): ROTATION RATE ABOUT Y AXIS
C X(3): ROTATION RATE ABOUT Z AXIS
C X(4): ROTATION ANGLE AROUND X AXIS
C A: ROTATION MATRIX
```

```
C
      A(1,1)=1.0D0
      A(1,2)=0.0D0
      A(1,3)=0.0D0
      A(2,1)=0.0D0
      A(2,2)=DCOS(X(4))
      A(2,3)=-1.0D0*DSIN(X(4))
      A(3,1)=0.0D0
      A(3,2)=DSIN(X(4))
      A(3,3)=DCOS(X(4))
C
      DO 10 I=1,3
      Y(I)=0.0D0
      DO 20 J=1,3
      Y(I)=Y(I)+X(J)*A(I,J)
 20   CONTINUE
 10   CONTINUE
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU10 >>>>>>>>>>>>>>>>>>>> ZZSU10 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU10(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
      IMPLICIT REAL*8(A-H,O-Z)
      IMPLICIT INTEGER(I-N)
      CHARACTER*72 DESC(20)
      LOGICAL DOF
      DIMENSION X(20),Y(20),P(20),A(3,3)
C
C***ZZSU10*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
       NUML= 2
       DESC(1)=' ZZSU10: TRANSFORMS THE RIGID BODY ROTATION RATES THROUGH'
       DESC(2)=' TRANSFORMATIONS ABOUT THE Y AND Z AXES.  THIS '
       DESC(3)=' SUBROUTINE IS USED IN FUNCTION BLOCK A20.'
C
      ELSE
C----- Evaluation section
C
C X(1): ROTATION RATE AROUND X AXIS
C X(2): ROTATION RATE AROUND Y AXIS
C X(3): ROTATION RATE AROUND Z AXIS
C X(4): ROTATION ANGLE ABOUT Y AXIS
C X(5): ROTATION ANGLE ABOUT Z AXIS
C A: ROTATION MATRIX
```



```
C
      A(1,1)=DCOS(X(5))*DCOS(X(4))
      A(1,2)=-1.0D0*DSIN(X(5))
      A(1,3)=DCOS(X(5))*DSIN(X(4))
      A(2,1)=DSIN(X(5))*DCOS(X(4))
      A(2,2)=DCOS(X(5));
      A(2,3)=DSIN(X(5))*DSIN(X(4))
      A(3,1)=-1.0D0*DSIN(X(4))
      A(3,2)=0.0D0
      A(3,3)=DCOS(X(4))
C
      DO 10 I=1,3
      Y(I)=0.0D0
      DO 20 J=1,3
      Y(I)=Y(I)+X(J)*A(I,J)
 20   CONTINUE
 10   CONTINUE
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU11 >>>>>>>>>>>>>>>>>>>> ZZSU11 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU11(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:  Use as a single-output-variable function.
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of the function
C             NUML,   number of lines in the description
C
C--- DECLARATIONS:
      IMPLICIT REAL*8(A-H,O-Z)
      IMPLICIT INTEGER(I-N)
      CHARACTER*72 DESC(20)
      LOGICAL DOF
      DIMENSION X(20),Y(20),P(20),A(3,3),YI(3),B(3,3)
C
C***ZZSU11*********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
       NUML= 2
       DESC(1)=' ZZSU11: CONVERTS THE SYSTEM-MISSILE VECTOR FROM'
       DESC(2)=' THE INERTIAL FRAME TO THE RIGID BODY FRAME.  THIS'
       DESC(3)=' IS USED INSIDE FUNCTION BLOCK A30.'
C
      ELSE
C----- Evaluation section
C
C X(1): X COORDINATE OF VECTOR
C X(2): Y COORDINATE OF VECTOR
C X(3): Z COORDINATE OF VECTOR
C X(4): ROTATION ANGLE ABOUT X AXIS
C X(5): ROTATION ANGLE ABOUT Y AXIS
```

```
C X(6): ROTATION ANGLE ABOUT Z AXIS
C A: ROTATION MATRIX FROM EARTH TO RIGID BODY FRAME
C
      A(1,1)=DCOS(X(6))*DCOS(X(5))
      A(1,2)=DSIN(X(6))*DCOS(X(4))+DCOS(X(6))*DSIN(X(5))*DSIN(X(4))
      A(1,3)=DSIN(X(6))*DSIN(X(4))-DCOS(X(6))*DSIN(X(5))*DCOS(X(4))
      A(2,1)=-1.0D0*DSIN(X(6))*DCOS(X(5))
      A(2,2)=DCOS(X(6))*DCOS(X(4))-DSIN(X(6))*DSIN(X(5))*DSIN(X(4))
      A(2,3)=DSIN(X(6))*DSIN(X(5))*DCOS(X(4))+DCOS(X(6))*DSIN(X(4))
      A(3,1)=DSIN(X(5))
      A(3,2)=-1.0D0*DCOS(X(5))*DSIN(X(4))
      A(3,3)=DCOS(X(5))*DCOS(X(4))
C
      DO 10 I=1,3
      YI(I)=0.0D0
      DO 20 J=1,3
      YI(I)=YI(I)+X(J)*A(I,J)
 20   CONTINUE
 10   CONTINUE
C
      Y(1)=YI(1)
      Y(2)=YI(2)
      Y(3)=YI(3)
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU12 >>>>>>>>>>>>>>>>>>>> ZZSU12 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU12(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       COMMON/EIGEN/EV,EVT
       COMMON/DFT/DFTD,NSAVT,DATA
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),EV(192,5),EVT(5,192),CA(3),CAD(3),
     +CV(3),AV(3),AVR(3),A(3,3),B(3,3),XR(3),RA(3),HOLD(4),
     +DFTD(4096,4)
C
C***ZZSU12**********************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 4
        DESC(1)='ZZSU12: CALCULATES THE POSITION OF THE MIRROR'
        DESC(2)='WITH RESPECT TO THE RIGID BODY AND THEN CONVERTS'
```



```
        DESC(3)='THE SYSTEM-MISSILE VECTOR FROM RIGID BODY FRAME'
        DESC(4)='TO THE MIRROR FRAME FOR INPUT TO THE OPTICS ROUTINE.'

C
      ELSE
C----- Evaluation section
C
C X(1): X COORDINATE OF SYSTEM-TARGET VECTOR IN EARTH FRAME
C X(2): Y COORDINATE OF SYSTEM-TARGET VECTOR IN EARTH FRAME
C X(3): Z COORDINATE OF SYSTEM-TARGET VECTOR IN EARTH FRAME
C X(4): TIME
C X(5): MODAL POSITION #1
C X(6): MODAL POSITION #2
C X(7): MODAL POSITION #3
C X(8): MODAL POSITION #4
C X(9): MODAL POSITION #5
C Y(1): VECTOR AZIMUTH WRT MIRROR Z AXIS
C Y(2): VECTOR ELEVATION WRT MIRROR Z AXIS
C Y(3): VECTOR AZIMUTH WRT RIGID BODY Z AXIS
C Y(4): VECTOR ELEVATION WRT RIGID BODY Z AXIS
C P(1): X COMPONENT OF TRANSLATION VECTOR FROM RIGID BODY CENTER TO MIRROR
C P(2): Y COMPONENT OF TRANSLATION VECTOR FROM RIGID BODY CENTER TO MIRROR
C P(3): Z COMPONENT OF TRANSLATION VECTOR FROM RIGID BODY CENTER TO MIRROR
C P(4): CONTROL AUTHORITY FLAG; 0:CONTROL FROM RIGID BODY ANGLES
C       1: CONTROL FROM MIRROR ANGLES
C VMAG: VECTOR MAGNITUDE
C SYSTEM2.F06: FILE WITH MODAL MATRIX INSIDE
C RA(I): ROTATION OF MIRROR CONNECTION POINT WRT TO RIGID BODY AXES
C
C IF TIME=0, THEN READ IN EIGENVECTORS
C
      VMAG=DSQRT(X(1)**2+X(2)**2+X(3)**2)
      PMAG=DSQRT(X(1)**2+X(3)**2)
      HOLD(1)=DASIN(X(1)/PMAG)
      HOLD(2)=DASIN(X(2)/VMAG)
      IF (X(4) .EQ. 0.0D0)THEN
      OPEN(4, FILE='SYSTEM2.F06',STATUS='OLD')
      DO 10 J=1,5
      READ(4,100)
      DO 10 K=1,187,6
 10   READ(4,200)(EV(I,J),I=K,K+5)
      CLOSE(4)
      DO 20 I=1,192
      DO 20 J=1,5
 20   EVT(J,I)=EV(I,J)
 100  FORMAT(1X,//////)
 200  FORMAT(26X,6(D13.7,2X))
      ENDIF
C
C CALCULATE GENERALIZED COORDINATE POSITIONS FROM MODAL POSTIONS
C
      DO 30 I=1,3
      CAD(I)=0.0D0
      RA(I)=0.0D0
      DO 40 J=1,5
      CAD(I)=CAD(I)+EV(I+24,J)*X(J+4)
      RA(I)=RA(I)+EV(I+27,J)*X(J+4)
 40   CONTINUE
 30   CONTINUE
C
C SET UP ROTATION MATRICES
C
      A(1,1)=DCOS(RA(2))*DCOS(RA(3))
      A(1,2)=-DCOS(RA(2))*DSIN(RA(3))
      A(1,3)=DSIN(RA(2))
      A(2,1)=DSIN(RA(1))*DSIN(RA(2))*DCOS(RA(3))+DCOS(RA(1))*DSIN(RA(3))
```

```
      A(2,2)=-DSIN(RA(1))*DSIN(RA(2))*DSIN(RA(3))+DCOS(RA(1))
     +*DCOS(RA(3))
      A(2,3)=-DSIN(RA(1))*DCOS(RA(2))
      A(3,1)=-DCOS(RA(1))*DSIN(RA(2))*DCOS(RA(3))+DSIN(RA(1))
     +*DSIN(RA(3))
      A(3,2)=DCOS(RA(1))*DSIN(RA(2))*DSIN(RA(3))+DSIN(RA(1))*DCOS(RA(3))
      A(3,3)=DCOS(RA(1))*DCOS(RA(2))
C
      B(1,1)=DCOS(RA(2))*DCOS(RA(3))
      B(1,2)=DCOS(RA(2))*DSIN(RA(3))
      B(1,3)=-DSIN(RA(2))
      B(2,1)=DSIN(RA(1))*DSIN(RA(2))*DCOS(RA(3))-DCOS(RA(1))*DSIN(RA(3))
      B(2,2)=DSIN(RA(1))*DSIN(RA(2))*DSIN(RA(3))+DCOS(RA(1))*DCOS(RA(3))
      B(2,3)=DSIN(RA(1))*DCOS(RA(2))
      B(3,1)=DCOS(RA(1))*DSIN(RA(2))*DCOS(RA(3))+DSIN(RA(1))*DSIN(RA(3))
      B(3,2)=DCOS(RA(1))*DSIN(RA(2))*DSIN(RA(3))-DSIN(RA(1))*DCOS(RA(3))
      B(3,3)=DCOS(RA(1))*DCOS(RA(2))
C
C SET UP TRANSLATION VECTORS
C
C FROM STRUCTURE TO MIRROR VERTEX
C
      AV(1)=0.0D0
      AV(2)=-P(2)
      AV(3)=-0.03125D0
C
C FROM STRUCTURE CENTER OF MASS TO MIRROR CONNECTION
C
      CA(1)=P(1)
      CA(2)=P(2)
      CA(3)=P(3)
C
C COMPUTE ROTATED STRUCTURE TO VERTEX VECTOR AND ADD ALL VECTORS
C TO GET VERTEX POSITION WRT TO CENTER OF MASS.  THEN FIND NEW VECTOR
C FROM MIRROR TO TARGET.
C
      DO 50 I=1,3
      AVR(I)=0.0D0
      DO 60 J=1,3
      AVR(I)=AVR(I)+AV(J)*A(I,J)
 60   CONTINUE
      CV(I)=CA(I)+CAD(I)+AVR(I)
      XR(I)=X(I)-CV(I)
 50   CONTINUE
C
C ROTATE MIRROR TARGET VECTOR INTO MIRROR FRAME
C
      DO 70 I=1,3
      X(I)=0.0D0
      DO 80 J=1,3
      X(I)=X(I)+XR(J)*B(I,J)
 80   CONTINUE
 70   CONTINUE
C
C SAVE TARGET VECTOR FOR OPTICS PROGRAM
C
      Y(5)=X(1)
      Y(6)=X(2)
      Y(7)=X(3)
C
C CALCULATE ANGLES
C
      VMAG=DSQRT(X(1)**2+X(2)**2+X(3)**2)
      PMAG=DSQRT(X(1)**2+X(3)**2)
      HOLD(3)=DASIN(X(1)/PMAG)
      HOLD(4)=DASIN(X(2)/VMAG)
```



```
C
C OUTPUT CONTROL DATA
C
      IF(P(4) .EQ. 0.0D0) THEN
      Y(1)=HOLD(1)
      Y(2)=HOLD(2)
      Y(3)=HOLD(3)
      Y(4)=HOLD(4)
      ELSE
      Y(1)=HOLD(3)
      Y(2)=HOLD(4)
      Y(3)=HOLD(1)
      Y(4)=HOLD(2)
      ENDIF
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU13 >>>>>>>>>>>>>>>>>>>> ZZSU13 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU13(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of the function
C             NUML,   number of lines in the description
C
C--- DECLARATIONS:
      IMPLICIT REAL*8(A-H,O-Z)
      IMPLICIT INTEGER(I-N)
      CHARACTER*72 DESC(20)
      LOGICAL DOF
      DIMENSION X(20),Y(20),P(20),VPP(3)
C
C***ZZSU13*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
       NUML= 1
       DESC(1)=' ZZSU13: NOT USED'
C
      ELSE
C----- Evaluation section
C
       Y(1)=X(1)*P(1)
C     ENDIF
      RETURN
      END
C>>>>>
C
CZZSU14 >>>>>>>>>>>>>>>>>>>> ZZSU14 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU14(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
```

```
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU14*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 1
        DESC(1)=' ZZSU14: NOT USED'
C
      ELSE
C----- Evaluation section
       Y(1)=X(1)*P(1)
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU15 >>>>>>>>>>>>>>>>>>>> ZZSU15 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU15(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       COMMON/AZDAT/OAZP,OAZT,OAZR
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU15*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
```



```
C----- Description section (max 20 lines)
        NUML= 1
        DESC(1)=' ZZSU15: CALCULATES AZIMUTH ERROR RATE FROM '
        DESC(2)=' CONSEQUITIVE AZIMUTH ERROR POSITIONS DIVIDED BY '
        DESC(3)=' CHANGE IN TIME.'
C
      ELSE
C----- Evaluation section
C
C X(1): AZIMUTH POSITION ERROR
C X(2): TIME
C Y(1): AZIMUTH POSITION ERROR
C Y(2): AZIMUTH RATE ERROR
C OAZP: OLD AZIMUTH POSITION ERROR
C OAZT: OLD AZIMUTH CALCULATION TIME
C OAZR: OLD AZIMUTH RATE ERROR
C
      IF(X(2) .EQ. 0.0D0)THEN
      Y(1)=X(1)
      Y(2)=0.0D0
      OAZP=X(1)
      OAZT=X(2)
      OAZR=0.0D0
      RETURN
      ENDIF
      DELT=X(2)-OAZT
      IF(DELT .EQ. 0.0D0)THEN
      Y(1)=X(1)
      Y(2)=OAZR
      Y(3)=OAZP
      Y(4)=OAZT
      Y(5)=OAZR
      Y(6)=DELT
      ELSE
      Y(1)=X(1)
      Y(2)=(X(1)-OAZP)/DELT
      Y(3)=OAZP
      Y(4)=OAZT
      Y(5)=OAZR
      Y(6)=DELT
      OAZP=X(1)
      OAZT=X(2)
      OAZR=Y(2)
      ENDIF
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU16 >>>>>>>>>>>>>>>>>>>> ZZSU16 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU16(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
```

```
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       COMMON/ELDAT/OELP,OELT,OELR
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20)
C
C***ZZSU16*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 3
        DESC(1)=' ZZSU16: CALCULATES ERROR IN VELOCITY BETWEEN TARGET'
        DESC(2)=' AND SYSTEM FROM TWO POSITION ERROR VALUES.  ROUTINE'
        DESC(3)=' REQUIRES CHANGE IN TIME BETWEEN READINGS.'
C
      ELSE
C----- Evaluation section
C
C X(1): CURRENT POSITION ERROR
C X(2): CURRENT TIME
C Y(1): POSTION ERROR
C OELP: OLD ELEVATION ERROR POSITION
C OELT: OLD ELEVATION CALCULATION TIME
C OELR: OLD ELEVATION ERROR RATE
C
      IF(X(2) .EQ. 0.0D0)THEN
      Y(1)=X(1)
      Y(2)=0.0D0
      OELP=X(1)
      OELT=X(2)
      OELR=0.0D0
      RETURN
      ENDIF
      DELT=X(2)-OELT
      IF(DELT .EQ. 0.0D0)THEN
      Y(1)=X(1)
      Y(2)=OELR
      Y(3)=OELP
      Y(4)=OELT
      Y(5)=OELR
      Y(6)=DELT
      ELSE
      Y(1)=X(1)
      Y(2)=(X(1)-OELP)/DELT
      Y(3)=OELP
      Y(4)=OELT
      Y(5)=OELR
      Y(6)=DELT
      OELP=X(1)
      OELT=X(2)
      OELR=Y(2)
      ENDIF
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU17 >>>>>>>>>>>>>>>>>>>> ZZSU17 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU17(X,P,Y,DOF,NUML,DESC)
C
```



```
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
       COMMON/EIGEN/EV,EVT
       CHARACTER*72 DESC(20)
       LOGICAL DOF
       DIMENSION X(20),Y(20),P(20),EV(192,5),EVT(5,192)
C
C***ZZSU17*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
        NUML= 4
        DESC(1)=' ZZSU17: READS THE MODAL MATRIX U TO PROVIDE THE'
        DESC(2)=' TRANSFORMER MODULATOR FOR CONVERTING GENERALIZED'
        DESC(3)=' FORCES INTO MODAL FORCES FOR THE STRUCTURAL VIBRATIONS'
        DESC(4)=' MODAL.'
C
      ELSE
C----- Evaluation section
C
C P(1): FORCE INPUT NUMBER
C P(2): MODE NUMBER
C EVT: MODAL MATRIX U
C
        Y(1)=EVT(INT(P(2)),INT(P(1)))
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU18 >>>>>>>>>>>>>>>>>>>> ZZSU18 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
       SUBROUTINE ZZSU18(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
C             P,      vector of parameter values
C             DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C             DESC,   description of function
C             NUML,   number of lines in description
C
C--- DECLARATIONS:
       IMPLICIT REAL*8(A-H,O-Z)
       IMPLICIT INTEGER(I-N)
```

```
      COMMON/FILTER/DATA,NLASTT,SAVE
      CHARACTER*72 DESC(20)
      LOGICAL DOF
      DIMENSION X(20),Y(20),P(20),DATA(360,5)
C
C***ZZSU18*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
       NUML= 2
       DESC(1)=' ZZSU18: SAVES FIVE INPUT VARIABLES AT 1 SECOND INTERVALS'
       DESC(2)=' IN THE FILE ENOUT.DAT AT THE END OF 360 SECS SIMULATION TIME.'
C
C X(1): INPUT DATA #1
C X(2): INPUT DATA #2
C X(3): INPUT DATA #3
C X(4): INPUT DATA #4
C X(5): INPUT DATA #5
C X(6): TIME
C DATA: MATRIX OF SAVED DATA
C SAVE: SAVE DATA FLAG
C NLASTT: TIME OF LAST SAVED DATA
C
      ELSE
C----- Evaluation section
C
      Y(1)=X(5)
      IF(X(6) .EQ. 0.0D0)THEN
      SAVE=1.0D0
      NLASTT=-1.0D0
      ENDIF
      IF(SAVE .LT. 1.0D0 .OR. X(6) .LT. 1.0D0)RETURN
      NCURTT=INT(X(6))
      IF(NCURTT .EQ. NLASTT)RETURN
      NLASTT=NCURTT
      DATA(NCURTT,1)=X(1)
      DATA(NCURTT,2)=X(2)
      DATA(NCURTT,3)=X(3)
      DATA(NCURTT,4)=X(4)
      DATA(NCURTT,5)=X(5)
      IF(NCURTT .GT. 360)GOTO 300
      RETURN
 300  OPEN(5,FILE='ENOUT.DAT',STATUS='NEW')
      DO 310 I=1,360
 310  WRITE(5,*)I,(DATA(I,J),J=1,5)
      SAVE=0.0
      CLOSE(5)
 400  FORMAT(1X,I5,2X,4(F15.12,2X))
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
CZZSU19 >>>>>>>>>>>>>>>>>>>> ZZSU19 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
      SUBROUTINE ZZSU19(X,P,Y,DOF,NUML,DESC)
C
C--- PROGRAMMING:
C
C--- DESCRIPTION:
C
C--- INPUTS:  X,      vector of input argument values
```



```
C            P,      vector of parameter values
C            DOF,    if '.TRUE.' then returns description
C
C--- OUTPUTS: Y,      vector of output argument values
C            DESC,   description of function
C            NUML,   number of lines in description
C
C--- DECLARATIONS:
      IMPLICIT REAL*8(A-H,O-Z)
      IMPLICIT INTEGER(I-N)
      CHARACTER*72 DESC(20)
      LOGICAL DOF
      DIMENSION X(20),Y(20),P(20)
C
C***ZZSU19*****************************************************************
C
C---  For detailed instructions see file header.
C
      IF (DOF) THEN
C----- Description section (max 20 lines)
       NUML= 1
       DESC(1)=' ZZSU19: NOT SET.'
C
      ELSE
C----- Evaluation section
C
       Y(1)=P(1)*X(1)
C
      ENDIF
      RETURN
      END
C>>>>>
C
C
```

Bibliography

ACOSS III, 1980. ACOSS THREE (Active Control of Space Structures) Phase I, Lockheed Missiles and Space Company, Inc., Defense Advanced Research Projects Agency (DoD), RADC-TR-80-131, May 1980.

Agrawal, B. N., 1986. Agrawal, B. N.,*Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., 1986.

AFWL, 1985. Air Force Weapons Laboratory. Contract FY3592-86-10181 with Boeing Aerospace & Electronics. Kirtland AFB NM, 21 October 1985.

Bos and Breedveld, 1985. Bos, A. M. and Breedveld, P. C.,*1985 Update of the Bond Graph Bibliography*, Journal of the Franklin Institute, Vol 319, No.1/2, Jan/Feb 1985.

Box and Draper, 1987. *Empirical Model-Building and Response Surfaces* New York: McGraw-Hill Book Company, 1987.

Brewer, J., 1990. Brewer, Capt Jeb Ewell. *Statistical Techniques for Designing a Decoupled Controller to be Robust to Model and Sensor Noise.* MS Thesis, AFIT/GA/ENY/90M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1990.

Briggs and others, No Date. Briggs, Hugh C. and others. *Integrated Control/Structures/Optics Dynamic Performance Modeling of a Segmented Reflector Telescope.* Prepublished report. Jet Propulsion Laboratory, Pasadena, CA. No date.

D'Azzo and Houpis, 1988. D'Azzo, John J. and Constantine H. Houpis. *Linear Control System Analysis and Design.* New York: McGraw-Hill Book Company, 1988.

Gere, J. and Weaver, W. 1965. Gere, James M. and William Weaver, Jr. *Analysis of Framed Structures* New Jersey: D. Van Nostrand Company, INC. 1965.

Greenwood, 1965. Greenwood, Donald T. *Principles of Dynamics.* New York: Prentice-Hall, 1965.

Hall, A., 1969. Hall, Arthur D. III *Three-Dimensional Morphology of System Engineering.* IEEE Trans. Systems Science and Cybernetics, Vol SSC-5, No2, APR 1969 pp 156-160

Hardy, A. and Perrin F., 1932. Hardy, Arthur C. and Fred H. Perrin. *The Principles of Optics.* New York: McGraw Hill Book Company, 1932.

Hill and Warfield, 1972. Hill, Douglas J. and Warfield, John N. *Unified Program Planning* IEEE Trans. Systems, Man, and Cybernetics, Vol SMC-2 No. 5 November 1972

Jacques, D., 1989. Jacques, Capt David R. *Baseline Experiment for Active Control of Structural Vibrations.* MS Thesis, AFIT/GAE/ENY/89D-15. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.

Khuri, A. I. and Conlon, M., 1981. Khuri, A. I., and Conlon, M., "Simultaneous Optimization of Multiple Responses Represented by Polynomial Regression Functions," Technometrics, Vol 23, No. 4, Nov 1981.

Likins, P., 1973. Likins, Peter W. *Elements of Engineering Mechanics.* New York: McGraw Hill Book Company, 1973.

Macneal-Schwendler Corp., 1983. MacNeal-Schwendler Corporation. *MSC/NASTRAN version 63, Handbook for Dynamic Analysis.* California: MacNeal-Scwendler Corporation, 1983.

Margolis, D. L., 1980. Margolis, D. L., "Dynamical Models for Multidimensional Structures Using Bond Graphs," Journal of Dynamic Systems, Measurement, and Control, Vol 102, Sept 1980.

Margolis, D. L., 1985. Margolis, D. L., "A Survey of Bond Graph Modeling for Interacting Lumped and Distributed Systems," Journal of the Franklin Institute, Vol 319, No.1/2, Jan/Feb 1985.

Margolis and Young, 1977. Margolis, D. L. and Young, G. E., "Reduction of Models of Large Scale Lumped Structures Using Normal Modes and Bond Graphs," Journal of the Franklin Institute, Vol 304, No. 1, July 1977.

Martin, J., 1987. Martin, Joe. *A Free-Free Beam for the Study of Spacecraft Control.* MS Thesis, The Ohio State University, Columbus, OH, 1987.

Meirovitch, L., 1986. Meirovitch, Leonard *Elements of Vibration Analysis* McGraw-Hill, Inc. 1986.

Meirovitch, L., 1967. Meirovitch, Leonard *Analytical Methods in Vibrations* Macmillan Publishing Co., Inc., 1967.

Millar, R.A., 1977. Millar, R.A. and F.R. Vigneron. "Attitude Stability and Structural Flexibility in Roll and Yaw for the Communications Technology Satellite." *Dynamics and Control of Large Flexible Spacecraft.* 41-56. VPI and SU, 1977.

Myers, R. H., 1976. *Response Surface Methodology* No Publisher,Virginia Polytechnic Institute and State University, 1976.

NSSA, 1973. *U.S. Space Launch Systems* (U), Navy Space Systems Activity. Report No. NSSA-R-20-72-2. P.O. Box 92960, Worldway Postal Center, Los Angeles, California, 90009, March 1973.

Pacejka, H. 1985. Pacejka H. B. *Modeling Complex Vehicle Systems Using Bond Graphs* Journal of the Franklin Institute Vol 319, No. 1/2 pp 67-81, January/February 1985.

Paynter, H. M., 1961. Paynter, H. M. *Analysis and Design of Engineering Systems* M. I. T. Press, Cambridge, Mass., 1961.

Redding, D., 1989. Redding, D. and W. Breckenridge. *Optical Modeling for Dynamics and Control Analysis.* Interoffice Memorandum, Memo DCR-89-34. Jet Propulsion Laboratory, Pasadena, CA, 29 November 1989.

Reid, 1983. Reid, J. Gary. *Linear System Fundamentals.* New York: McGraw-Hill, 1983.

RCA, 1974. RCA Corporation. *Electro-Optics Handbook.* New Jersey, 1974.

Rosenberg, R. and Karnopp, D., 1983. Rosenberg, Ronald C. and Dean C. Karnopp. *Introduction to Physical Systems Dynamics* New York: McGraw Hill Book Company, 1983.

Rosenberg, R. and Karnopp, D., 1972. Rosenberg, Ronald C. and Dean C. Karnopp. "A Definition of the Bond Graph Language" *Journal of Dynamic Systems, Measurement, and Control* Vol 94 Series G #3 179-182 (September 1972).

Shannon and Wyant, 1979. Shannon, R. R., and Wyant, J. C., *Applied Optics and Optical Engineering*, Academic Press, 1979.

Shigley, 1977. Shigley, Joseph Edward. *Mechanical Engineering Design.* New York: McGraw-Hill, 1977.

Sky and Telescope Magazine, Apr 90. *Sky and Telescope Magazine* page 302.

Spindler and Hoyer, 1989. Spindler and Hoyer. *Precision Optics.* Federal Republic of Germany: Spindler and Hoyer, 1988.

Tsai, S. W., 1987. Tsai, S. W., *Composites Design, 3rd Edition*, Think Composites, 1987.

Tsai and Hahn, 1980. Tsai, S. W. and Hahn, A. T., *Introduction to Composite Materials*, Technomic Publishing Co., 1980.

Walker, Vos, Price and Brogren, 1984. Walker, W. J., Vos, R. G., Price, E. W., and Brogren, E. W.,"IAC Executive Summary," NASA CR-175196, Boeing Aerospace Company, February 1984.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1990	Master's Thesis

4. TITLE AND SUBTITLE
Multidisciplinary Modeling and Design of a Space Structure

5. FUNDING NUMBERS

6. AUTHOR(S)
Brian K. Cassiday, Capt, USAF; Stephen O. Gaines, Capt, USAF; Lawrence L. Gatschet, Capt, USAF; Mario N. Moya, Capt, USAF; John T. Tester, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSE/ENY/90D-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
WL/ARCB, Kirtland AFB NM 87117-6008

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A method is needed to simultaneously integrate several engineering disciplines into one complete model to determine the performance of the system as a whole during the design development. System engineering methodologies are particularly suited to this problem. These methods provide a structured approach to problem formulation and system parameter identification. Bond graphs are also well suited as bond graphs model power flow and energy relationships within subsystems, a characteristic inherent to all dynamic systems.

The intent of this research was to demonstrate the use of concurrent engineering theory to systematically model a large flexible space structure involving several engineering disciplines. The disciplines considered include: dynamics, controls, optics, structures, and heat transfer. A set of optimal solutions is presented and the results of this research are compared and contrasted with the results from classical design and modelling techniques.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Systems Engineering, Models, Space Systems, Optimization, Bond Graphs, Multidisciplinary Modeling, Theses. (SDW)	261
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL